REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 20, 1991	6/89 to 11/89

4. TITLE AND SUBTITLE

Probabilistic Interference in Restrictive Systems

5. FUNDING NUMBERS

TA - RR015-09-41
WU - 2830-0-1-(6.1)

6. AUTHOR(S)

James W. Gray III

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL Report 9315

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Chief Naval Research
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Some of the results presented in this report were previously published in the Proceedings of the 1990 IEEE Symposium on Security and Privacy, Oakland, California.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Probabilistic interference in nondeterministic machines can be exploited by trojan horses to reliably leak information to unauthorized users. This problem has been noted by other researchers but has not previously been addressed. We extend McCullough's restrictiveness to additionally prevent probabilistic interference. Then, to illustrate the use of our extension, we develop a nondeterministic system that solves a denial of service problem, and we use our definition to prove that the system is secure. Finally, we prove a limited composability result.

14. SUBJECT TERMS

Computer security
Covert channel detection
Formal methods
Noninterference

15. NUMBER OF PAGES

35

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18
298-102

CONTENTS

PROBABILISTIC INTERFERENCE IN RESTRICTIVE SYSTEMS

1. INTRODUCTION

The notion of noninterference was first introduced by Goguen and Meseguer [1,2] to formally specify and verify security properties. Their formalism allows a specifier to state properties of the form: "commands from the set A, issued by users in the set G, do not interfere with users in the set G'." Goguen and Meseguer showed that a variety of security policies (including label-based mandatory access controls and identity-based discretionary access controls) could be specified by using this formalism. In addition to their wide applicability, noninterference assertions capture our intuition of security properties very well. For these reasons, the noninterference formalization is very appealing as the basis for a general theory of security.

One problem with Goguen and Meseguer's original formulation of noninterference is that they modeled computer systems as *deterministic* state machines. As discussed in Ref. 3, many computer systems are nondeterministic and therefore cannot be accurately modeled as deterministic machines. Recognizing this, Sutherland [4] and later McCullough [3,5] modeled computer systems as nondeterministic state machines and defined security policies in terms of those models.

In accordance with the view that large, distributed, secure computer systems should be built by hooking up independently built and verified component systems, McCullough proved that his definition of security, called *restrictiveness*, is composable (i.e., by hooking up two or more restrictive systems, a composite system which is restrictive is produced).

Despite the advances made to date, culminating with McCullough's definition of restrictiveness, some problems remain. First, verifying that a system is restrictive does not show anything about covert timing channels. Specifically, high events can interfere with the timing of low events (e.g., response time). This timing interference can be exploited by trojan horses to leak sensitive information to unauthorized users. In current practice, covert timing channel analyses are performed to find and determine the threat associated with these channels.

Second, verifying that a system is restrictive does not show anything about probabilistic channels; high events can interfere with the *probability* that a low event will occur. As with timing interference, probabilistic interference can be exploited by a trojan horse to reliably leak high information to unauthorized users. This problem has been noted by other researchers [5,6] but has not previously been addressed.

Third, for the types of interference that are prevented by restrictiveness, the policy cannot be relaxed to allow a small amount of interference. It has been said that computer systems "are often not intended to be completely secure" [7] and that any "real system will have channels that violate the noninterference policy" [6]. For example, low-bandwidth covert channels may be permitted for the sake of performance. For this reason, restrictiveness may be too strong a property for a real system to satisfy. In both Refs. 6 and 7 recommendations are made to partially address this problem.

On the one hand, restrictiveness does not prevent all types of interference (viz., timing and probabilistic interference) and therefore should be strengthened; on the other hand, restrictiveness is too inflexible to allow a small (i.e., somehow quantified and deemed to be sufficiently small) amount of insecurity and therefore should be weakened.

The ultimate objective of our research is to define a security property that completely captures the notion of noninterference (i.e., there are no loopholes like covert timing channels that must be addressed separately), and at the same time can be relaxed to allow some quantifiable amount of interference. Furthermore, this security property must be defined in terms of a sufficiently general system model (i.e., aspects

Manuscript approved April 27, 1990.

of real systems such as nondeterminacy must be representable in the model). We hope that such a property could be realistically applied in the development of a secure system to gain assurance that the system provides a specified level of protection.

Our long-term approach for achieving this objective is as follows. Our first objective is to define perfect noninterference. By perfect noninterference we mean that a system that is shown to be perfectly noninterfering cannot exhibit undesirable interference of any kind. We believe that only after we fully understand what it means for a system to be perfectly secure, we can properly define our tolerance for insecurity. Thus, our second objective will be to generalize perfect noninterference to allow a quantifiable amount (e.g., 22 bits/min) of interference.

It is toward the first objective—defining perfect noninterference—that the present work is aimed. In this report, we develop an extension to McCullough's restrictiveness that precludes probabilistic interference. In this report we also restate McCullough's state-machine formalism and definition of restrictiveness; we present an example system that illustrates the problem of probabilistic interference. Then we develop an extension to McCullough's work that solves the problem of probabilistic interference. We present a series of examples designed to show the application of our extension. and an example of a new solution to the so-called *secure readers-writers problem* [8]. At the end we discuss the composability of our extension, and we present our conclusions and plans for future work.

2. RESTRICTIVENESS

In Ref. 5 state machine restrictiveness is formalized in the following way:

Definition: A *state machine* Σ is given by a six tuple $(S, \sigma_0, E, I, O, T)$, where S is the set of all possible states, $\sigma_0 \in S$ is the initial state, E is the set of possible events, $I \subseteq E$ is the set of all input events, $O \subseteq E$ is the set of all output events, and $T \subseteq S \times E \times S$ is the set of all possible state transitions.

Definition: Extended transitions are given by $ET \subseteq S \times E^* \times S$ where $(\sigma_1, \langle e_1, \ldots, e_{n-1}\rangle, \sigma_n) \in ET$ if and only if some sequence of states $\sigma_2, \ldots, \sigma_{n-1}$ exists, such that $(\sigma_i, e_i, \sigma_{i+1}) \in T$ for all i, $0 < i < n$.

Definition: Let $\approx$ be an equivalence relation on states of a system Σ (specifying which states appear to be the same state from the point of view of a particular user) and v be a subset of E (specifying which events of Σ are visible to that user). We call $\langle v, \approx\rangle$ a *projection* of the system Σ.

The following condition for restrictiveness is exactly the same as McCullough's, restated in a more compact form. The condition that must be satisfied for a given projection to be restrictive is stated in two parts. Intuitively, part (1) says that invisible inputs do not affect the visible part of the state; part (2) says that the invisible part of the state does not affect whether or not visible events occur.

Definition: The projection $\langle v, \approx\rangle$ is *restrictive* for Σ if the following condition holds.

Let (σ_1, x, σ_1') be an arbitrary transition of Σ.

(1) $x \in I - v \Rightarrow \sigma_1 \approx \sigma_1'$ and

(2) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow (\exists \sigma_2' \in S)(\exists \gamma \in E^*)$

$[$(2a) $(\sigma_2, \gamma, \sigma_2') \in ET$,
(2b) $\sigma_2' \approx \sigma_1'$,
(2c) $x \in I \Rightarrow \gamma = \langle x\rangle$,
(2d) $x \in ((E - I) - v) \Rightarrow \gamma \in ((E - I) - v)^*$, and
(2e) $x \in ((E - I) \cap v) \Rightarrow (\exists \gamma_1, \gamma_2 \in ((E - I) - v)^*)[\gamma = \gamma_1 \hat{} \langle x\rangle \hat{} \gamma_2]]$.

Although McCullough does not give an "unwinding theorem", this condition is analagous to the unwound versions of noninterference given in Refs. 2 and 6.

3. PROBABILISTIC INTERFERENCE

In the previous definition, (2) intuitively says that the invisible part of the state does not interfere with whether or not a particular visible event can occur. However, it does not say that the invisible part of the state does not interfere with the probability with which a particular visible event will occur. For example, consider the following system that keeps track (via its internal state) of the most recent input, and from any state nondeterministically outputs either $Out0$ or $Out1$.

Let $\Sigma 1$ be the state machine given by $(S, \sigma_0, E, I, O, T)$, where

$$\begin{aligned} S &= \{0,1\} \\ \sigma_0 &= 0 \\ E &= \{In0, In1, Out0, Out1\} \\ I &= \{In0, In1\} \\ O &= \{Out0, Out1\} \\ T &= \{(0, In0, 0), (0, In1, 1), (0, Out0, 0), (0, Out1, 0), (1, In0, 0), (1, In1, 1), (1, Out0, 1), (1, Out1, 1)\}. \end{aligned}$$

According to the definition of T, in either state 0 or 1 the system can nondeterministically output $Out0$ or $Out1$. However, suppose that when an output occurs in state 0, 95% of the time it is $Out0$, and only 5% of the time it is $Out1$. And when an output occurs in state 1, 95% of the time it is $Out1$, and only 5% of the time it is $Out0$. These probabilities cannot be represented in McCullough's formalism; therefore, they do not affect whether or not the system is restrictive.

Theorem 1: Define the equivalence relation $\approx$ by $\sigma_1 \approx \sigma_2$ for all states, σ_1 and σ_2 (i.e., the user cannot distinguish state 0 from state 1). Let $v = \{Out0, Out1\}$ (i.e., the user can see outputs but not inputs). The projection $\langle v, \approx \rangle$ is restrictive for $\Sigma 1$.

Proof: Let (σ_1, x, σ_1') be an arbitrary transition of $\Sigma 1$.

Since $\sigma_1 \approx \sigma_2$ for all σ_1 and σ_2,

$$(1)\ \ x \in I - v \Rightarrow \sigma_1 \approx \sigma_1'$$

is trivially true.

Let σ_2 be an arbitrary state such that $\sigma_1 \approx \sigma_2$. We must show that

$$(2)\ \ (\exists \sigma_2' \in S)(\exists \gamma \in S^*)$$

$$\begin{aligned} &[(2a)\ \ (\sigma_2, \gamma, \sigma_2') \in ET, \\ &(2b)\ \ \sigma_2' \approx \sigma_1', \\ &(2c)\ \ x \in I \Rightarrow \gamma = \langle x \rangle, \\ &(2d)\ \ x \in ((E-I)-v) \Rightarrow \gamma \in ((E-I)-v)^*, \\ &(2e)\ \ x \in ((E-I) \cap v) \Rightarrow (\exists \gamma_1, \gamma_2 \in ((E-I)-v)^*)[\gamma = \gamma_1 \hat{} \langle x \rangle \hat{} \gamma_2]]. \end{aligned}$$

There are four cases.

Case 1: $x = In0$. Choose $\sigma_2' = 0$ and $\gamma = \langle In0 \rangle$. Then (2a) $[(\sigma_2, \gamma, \sigma_2') \in ET]$ holds, since in either state $In0$ may be received, after which the state will be 0; (2b) $(\sigma_2' \approx \sigma_1')$ holds since $\sigma_1 \approx \sigma_2$ for all σ_1 and σ_2; (2c) $x \in I \Rightarrow \gamma = \langle x \rangle$ holds since $\gamma = \langle In0 \rangle = \langle x \rangle$; and (2d) and (2e) hold trivially since $x \notin (E-I)$.

Case 2: $x = In1$. Choose $\sigma_2' = 1$ and $\gamma = \langle In1 \rangle$. Then (2a)–(2e) all hold by similar arguments.

Case 3: $x = Out0$. Choose $\sigma_2' = \sigma_2$ and $\gamma = \langle Out0 \rangle$. We have two subcases.

Case 3.1: $\sigma_2 = 0$. In this case, 95% of the time, $Out0$ will be output, so $(\sigma_2, \gamma, \sigma_2') \in ET$ is true.

Case 3.2: $\sigma_2 = 1$. In this case, 5% of the time, $Out0$ will be output, so $(\sigma_2, \gamma, \sigma_2') \in ET$ is true. Therefore, (2a) holds; (2b) again holds since $\sigma_1 \approx \sigma_2$ for all σ_1 and σ_2; (2c) and (2e) hold trivially since $x \notin I$ and $x \notin v$; (2d) holds since $\gamma = \langle Out0 \rangle = \langle x \rangle$ and $x \in ((E-I)-v) \Rightarrow \langle x \rangle \in ((E-I)-v)^*$.

Case 4: $x = Out1$. Choose $\sigma_2' = \sigma_2$ and $\gamma = \langle Out1 \rangle$. Then (2a)–(2e) hold by similar arguments.

Thus, $\langle v, \approx \rangle$ is restrictive for $\Sigma 1$. []

We would like this theorem and proof to show that the inputs $In0$ and $In1$ do not interfere with the outputs $Out0$ and $Out1$. However, 95% of the time the outputs accurately convey which input was the most recent one.

What the theorem actually says is that the inputs $In0$ and $In1$ interfere only with the invisible part of the system state, and that the invisible part of the state does not interfere with whether or not visible events can occur. The security problem arises because the invisible part of the state does interfere with the probability with which visible events occur. Thus, a noisy but potentially dangerous (and potentially high bandwidth) channel can exist in a system that is shown to be restrictive. We call this problem probabilistic interference. McCullough [3,9] gives examples of probabilistic interference to illustrate that deducibility security [Sutherland 86] does not rule out all insecure systems. McCullough also states that restrictiveness "disallows all kinds of definite channels (ones that don't involve probabilistic inferences),"[5] where "probabilistic inferences" appears to mean what we term probabilistic interference. The problem has also been noted in Ref. 6, where they ignored nondeterminism and thus did not address the problem.

4. FORMALIZING THE PROBABILITY OF EVENTS

In this section we incorporate probabilistic concerns into the treatment of state machines and restrictiveness, and then reconsider $\Sigma 1$, the example system from the previous section.

4.1. State Machines

We modify McCullough's formalization of state machines as follows.

A state machine Σ is given by a six tuple $(S, \sigma_0, E, I, O, T)$, where S is the set of all possible states, σ_0 is the initial state, E is the set of possible events, $I \subseteq E$ is the set of all input events, $O \subseteq E$ is the set of all output events, and $T \subseteq S \times E. \times S \times [0,1]$ is the set of all possible state transitions.

The meaning of $(\sigma_1, e, \sigma_2, p) \in T$ is as follows:

- If $e \in E - I$, then whenever the system is in state σ_1, the system will engage in e and transition to σ_2 with probability p.
- If $e \in I$ then whenever the system is in state σ_1, the system will, with probability p, *attempt* to accept e and transition to σ_2. If the environment is not offering e (e.g., a user has not entered e), then on this attempt the system will perform the null transition (i.e., the system will transistion from σ_1 to σ_1 without engaging in any visible event).

This action of a system attempting to accept an input can be thought of as polling: The system checks whether the environment is ready to provide the input: if the environment is ready, then the system accepts the input and makes its transition; if not, then the system does nothing.

This method of obtaining input can hinder good system performance (e.g., due to busy waiting), therefore, for performance purposes the preferred method of obtaining input is with interrupts. However for our purpose of preventing interference, interrupts can cause problems. For example, if a high subject can interrupt a system that interacts with a low subject, the high subject can interfere (probabilistically and/or temporally) with the low subject by varying the frequency of its interrupts. By using the polling

method of obtaining inputs, a system controls when it will accept an input and thus has complete control over whether high inputs interfere with low outputs. For this reason, we chose to include only the polling method of obtaining input in our system model.

Another effect of the polling method is that it is no longer necessary for systems to be input total (i.e., a system can decide not to accept an input and the input may be lost). Therefore, in this report we do not require that systems be input total. Thus, there are systems (which are not input total) that are not restrictive but do satisfy our definition of security.

Even though the polling method of obtaining inputs is more suitable for security purposes, cases exist where interrupts are useful and do not cause security problems (e.g., a user interface that interacts with a single user at a single security level could be driven by interrupts from the keyboard), therefore a fully general system model should include facilities for specifying and reasoning about interrupts.

Note: For the probabilities of events to make sense, the sum of the probabilities of all next possible events should equal 1. However, for security purposes, we do not need to make this requirement on systems. We consider feasibility for implementation to be a separate issue from security. Thus, a specification of a system may be shown to be secure and at the same time be impossible to implement as specified.

4.2. P-Restrictiveness

In this section we incorporate constraints on probabilistic interference into McCullough's state machine restrictiveness. First we formalize the probability that the system, starting in state σ_1, will (with respect to the projection $\langle v, \approx \rangle$) appear to engage in the event x and transition to state σ_2.

Definition: Let

$$p_{(\sigma_1,x,\sigma_2)} = \begin{cases} p \text{ such that } (\sigma_1, x, \sigma_2, p) \in T, & \text{if such a } p \text{ exists;} \\ 0, & \text{otherwise.} \end{cases}$$

Now, for a given projection $\langle v, \approx \rangle$, define $P_{\langle v,\approx\rangle} : S \times E \times S \to [0,1]$ as

$$P_{\langle v,\approx\rangle}(\sigma_1, x, \sigma_2) = \begin{cases} \sum\limits_{\sigma_2' \approx \sigma_2} p_{(\sigma_1,x,\sigma_2')} & \text{if } x \in v; \\ \sum\limits_{x' \in E-v \text{ and } \sigma_2' \approx \sigma_2} p_{(\sigma_1,x',\sigma_2')} & \text{if } x \notin v. \end{cases}$$

This definition is an integral part of the definition of P-restrictiveness, and so we would like to point out a few subtleties.

First, note that the probabilities of all transitions from σ_1 (i.e., only σ_1) to any state equivalent to σ_2 are summed. This means that $P_{\langle v,\approx\rangle}(\sigma_1, x, \sigma_2)$ is the probability that the system will, from σ_1, transition on x (or any invisible event if x is invisible) to a state equivalent to σ_2. The reason for defining $P_{\langle v,\approx\rangle}$ this way (rather than as the probability that the system will, from any state equivalent to σ_1, transition on ...) should be clear after the definition of P-restrictiveness has been presented.

Second, note that for an invisible event x, the summation includes transitions on any invisible event. This is because from the point of view of the projection $\langle v, \approx \rangle$, any two transitions from σ_1 to equivalent (with respect to $\approx$) states, that engage in invisible (with respect to v) events will appear to be the same.

Third, note that the second case applies for all $x \notin v$. This means that for an x that is not in E (i.e., not even a possible event of the system), $P_{\langle v,\approx\rangle}(\sigma_1, x, \sigma_2)$ may be positive. Again this is due to the point of view of the projection $\langle v, \approx \rangle$. To a user with projection $\langle v, \approx \rangle$, a possible system event that is not in v and another event that is not even a possible system event will appear the same—they are both invisible.

Now we present our extension to McCullough's definition of restrictiveness.

Definition: Let $\approx$ be an equivalence relation on states of a system Σ, and v be a subset of E. The projection $\langle v, \approx \rangle$ is *probability-extended-restrictive* (P-restrictive) if the following condition holds.

Let $\sigma_1, \sigma_1' \in S$ be arbitrary states, $x \in E$ be an arbitrary event, and $p \in (0, 1]$ be a nonzero probability. $(\sigma_1, x, \sigma_1', p) \in T$ implies

(1) $x \in I - v \Rightarrow \sigma_1 \approx \sigma_1'$, and

$P_{\langle v, \approx \rangle}(\sigma_1, x, \sigma_1') = p$ implies

(2) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow (\exists \sigma_2' \in S)(\exists y \in E)$,

$[$(2a) $P_{\langle v, \approx \rangle}(\sigma_2, y, \sigma_2') = p$,
(2b) $\sigma_2' \approx \sigma_1'$,
(2c) $x \in I \Rightarrow y = x$,
(2d) $x \in ((E - I) - v) \Rightarrow y \in ((E - I) - v)$, and
(2e) $x \in ((E - I) \cap v) \Rightarrow y = x]$.

We made this initial statement of P-restrictiveness to emphasize its similarities and differences with McCullough's definition of restrictiveness. The differences are:

- The antecedent of (1) is changed from $(\sigma_1, x, \sigma_1') \in T$ to $(\sigma_1, x, \sigma_1', p) \in T$. This extension corresponds to the extension of the state machine formalization.
- In the antecedent of (2) and within (2a), $(\sigma, x, \sigma') \in T$ is changed to $P_{\langle v, \approx \rangle}(\sigma, x, \sigma') = p$. This modification represents the addition of constraints on the probabilities with which events occur.
- Within (2), the event sequence γ is changed to the event y (e.g., there is a loss of transitive closure in (2d)). The motivation for this change is to simplify the statement and application of P-restrictiveness (viz., we avoid computing the probability of the occurrence of arbitrarily long sequences of events and avoid computing the sum of infinite sets of probabilities of event sequences). This modification has the unfortunate consequence that some systems that are restrictive and that do *not* contain any probabilistic interference are not P-restrictive (i.e., P-restrictiveness excludes more systems from the set of all restrictive systems than just the ones that exhibit probabilistic interference). In section 5, we further extend our state machine model and definition of P-restrictiveness, which somewhat alleviates this problem.

Largely because of the subtleties of the definition of $P_{\langle v, \approx \rangle}$, this condition for P-restrictiveness can be restated in the following logically equivalent but simpler form.

Theorem 2: Let $\approx$ be an equivalence relation on states of a system Σ and v be a subset of E. The projection $\langle v, \approx \rangle$ is P-restrictive if the following condition holds.

Let $\sigma_1, \sigma_1' \in S$ be arbitrary states, $x \in E$ be an arbitrary event, and $p \in (0, 1]$ be a nonzero probability.

(1) $(\sigma_1, x, \sigma_1', p) \in T$ and $x \in I - v \Rightarrow \sigma_1 \approx \sigma_1'$ and

(2) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow P_{\langle v, \approx \rangle}(\sigma_1, x, \sigma_1') = P_{\langle v, \approx \rangle}(\sigma_2, x, \sigma_1')$.

Proof: Assume that for any states σ_1 and $\sigma_1' \in S$, any event $x \in E$, and any nonzero probability $p \in (0,1]$,

(1) $(\sigma_1, x, \sigma_1', p) \in T$ and $x \in I - v \Rightarrow \sigma_1 \approx \sigma_1'$ and

(2) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow P_{\langle v,\approx\rangle}(\sigma_1, x, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, x, \sigma_1')$.

We must show that the following holds:

Let $\sigma_1, \sigma_1' \in S$ be arbitrary states, $x \in E$ be an arbitrary event, and $p \in (0,1]$ be a nonzero probability.

$(\sigma_1, x, \sigma_1', p) \in T$ implies

(1′) $x \in I - v \Rightarrow \sigma_1 \approx \sigma_1'$ and

$P_{\langle v,\approx\rangle}(\sigma_1, x, \sigma_1') = p$ implies

(2′) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow (\exists \sigma_2' \in S)(\exists y \in E)$

$[$(2a′) $P_{\langle v,\approx\rangle}(\sigma_2, y, \sigma_2') = p$,
(2b′) $\sigma_2' \approx \sigma_1'$,
(2c′) $x \in I \Rightarrow y = x$,
(2d′) $x \in ((E - I) - v) \Rightarrow y \in ((E - I) - v)$, and
(2e′) $x \in ((E - I) \cap v) \Rightarrow y = x]$.

1′ follows directly from 1. By choosing $\sigma_2' = \sigma_1'$ and $y = x$, 2a′ through 2e′ follow directly from 2. []

Demonstrating that the original condition for P-restrictiveness (as stated in the definition) implies the condition in theorem 2 (i.e., demonstrating that the two conditions are in fact logically equivalent) requires the use of the definition of $P_{\langle v,\approx\rangle}$, but it is also straightforward. The simplified condition for P-restrictiveness given in theorem 2 (in addition to being easier to understand) makes the proof of P-restrictiveness easier.

4.3. Σ1 Reconsidered

In the probability extended state machine formalization of the previous section, Σ1 can be defined by $(S, \sigma_0, E, I, O, T)$, where

$S = \{0, 1\}$,
$\sigma_0 = 0$,
$E = \{In0, In1, Out0, Out1\}$,
$I = \{In0, In1\}$,
$O = \{Out0, Out1\}$, and
$T = \{(0, In0, 0, .25), (0, In1, 1, .25), (0, Out0, 0, .475), (0, Out1, 0, .025), (1, In0, 0, .25), (1, In1, 1, .25), (1, Out0, 1, .025), (1, Out1, 1, .475)\}$.

Theorem 3: Let $v = \{Out0, Out1\}$. There does not exist an equivalence relation, $\approx$ on states of Σ1, such that the projection $\langle v, \approx\rangle$ is P-restrictive for Σ1.

Proof: Since the occurrence of $In0$ and $In1$ can change the state of the system from 1 to 0 and from 0 to 1, respectively, and $In0$ and $In1$ are not members of v, for (1) to hold, the equivalence relation $\approx$ must be defined by $\sigma_1 \approx \sigma_2$ for all σ_1 and $\sigma_2 \in S$.

Therefore we only need to show that given $\approx$ is defined by $\sigma_1 \approx \sigma_2$ for all σ_1 and $\sigma_2 \in S$, $\langle v, \approx\rangle$ is not P-restrictive.

By the definition of P,

$$P_{\langle v,\approx\rangle}(0, Out0, 0) = \sum_{\sigma_2' \approx 0} p_{(0,Out0,\sigma_2')}$$

Since 0 is the only state σ_2' such that $p_{(0,Out0,\sigma_2')}$ is nonzero, and $p_{(0,Out0,\sigma_2')} = .475$,

$$P_{\langle v,\approx\rangle}(0, Out0, 0) = .475$$

Also by the definition of P,

$$P_{\langle v,\approx\rangle}(1, Out0, 0) = \sum_{\sigma_2' \approx 1} p_{(1,Out0,\sigma_2')}$$

Since 1 is the only state σ_2' such that $p_{(1,Out0,\sigma_2')}$ is nonzero, and $p_{(0,Out0,\sigma_2')} = .025$,

$$P_{\langle v,\approx\rangle}(1, Out0, 0) = .025$$

Since $0 \approx 1$, and $P_{\langle v,\approx\rangle}(0, Out0, 0) = .475 \neq .025 = P_{\langle v,\approx\rangle}(1, Out0, 0)$, $\langle v, \approx\rangle$ cannot be P-restricti for $\Sigma 1$. []

5. DENIAL OF SERVICE

This section presents an example of how nondeterminism can be used to prevent denial of servic First, a denial of service problem is given. A restrictive solution is presented that contains a probabilist covert channel and is not P-restrictive. Then, an alternative solution is presented that prevents denial service and is also P-restrictive.

By this series of examples, we hope to show:

(1) Systems that may appear to be reasonable and are restrictive, can contain probabilistic cove channels.
(2) A useful, nondeterministic system can be shown to be P-restrictive.
(3) Nondeterminism can be used to prevent denial of service without introducing insecurities.

5.1 The Secure Readers-Writers Problem

Consider the following simplified version of the secure readers-writers problem [8]. A single proce controls access to a single object. There are two users called "hi" and "lo". User hi wants to issue sequenc of commands of the form "begin read", "read", "read", ... "read", "end read". User lo wants to issu sequences of commands of the form "begin write", "write ⟨Object⟩", "write ⟨Object⟩", ..., "write ⟨Object⟩ "end write." (where ⟨Object⟩ is the value to be written to the controlled object). The integrity requiremer is: If the controlled object is modified (with a successfully executed "write ⟨Object⟩" command) sometir during a "begin read", "read", "read", ... "read", "end read" sequence then user hi must be notified. In th way, user hi will be alerted that the object may not have been in a consistent state during the sequence reads and may retry the sequence. The security requirement for this problem is that commands issued b hi may not interfere with the outputs seen by lo.

Note: This problem has been simplified from the general readers-writers problem (as it appeared i Ref. 8) in two ways:

(1) in the general problem there is more than one object, and
(2) in the general problem there are more than two users. In particular there may be more than one writer, and so there would be an additional integrity requirement to prevent more than one current writer.

5.2 A Small Modification to the Model

Before presenting solutions to the secure readers-writers problem, there is an extension to our model of state machines that we wish to make.

A state machine Σ is given by a six tuple $(S, \sigma_0, E, I, O, T)$ where S is the set of all possible states, σ_0 is the initial state, E is the set of possible events, $I \subseteq E$ is the set of all input events, $O \subseteq E$ is the set of all output events, and $T \subseteq S \times E^* \times S \times [0,1]$ is the set of all possible state transitions.

Definition: Let

$$p_{(\sigma_1,\gamma,\sigma_2)} = \begin{cases} p \text{ such that } (\sigma_1,\gamma,\sigma_2,p) \in T, & \text{if such a } p \text{ exists;} \\ 0, & \text{otherwise.} \end{cases}$$

Now, for a given projection $\langle v, \approx \rangle$, define $P_{\langle v,\approx\rangle} : S \times E^* \times S \to [0,1]$ as,

$$P_{\langle v,\approx\rangle}(\sigma_1,\gamma,\sigma_2) = \begin{cases} \sum\limits_{\sigma_2' \approx \sigma_2} p_{(\sigma_1,\gamma,\sigma_2')} & \text{if } \gamma \in v; \\ \sum\limits_{\substack{\gamma' \in E-v \text{ and} \\ \sigma_2' \approx \sigma_2}} p_{(\sigma_1,\gamma',\sigma_2')} & \text{if } \gamma \notin v. \end{cases}$$

Definition: The infix function $| : E^* \times \wp(E) \to E^*$ (called restriction), where $\wp(E)$ is the powerset of E, is defined recursively as follows: For any set of events $E1 \subseteq E$,

$$\langle\rangle \mid E1 = \langle\rangle$$

and for any $x \in E$ and any $\gamma \in E^*$,

$$(\langle x \rangle^\wedge \gamma) \mid E1 = \begin{cases} \gamma \mid E1 & \text{if } x \in E1; \\ \langle x \rangle^\wedge (\gamma \mid E1) & \text{otherwise.} \end{cases}$$

Definition: Let $\approx$ be an equivalence relation on states of a system Σ and v be a subset of E^*. The projection $\langle v, \approx \rangle$ is P-restrictive if the following condition holds.

Let $\sigma_1, \sigma_1' \in S$ be arbitrary states, $\gamma \in E^*$ be an arbitrary event sequence, and $p \in (0,1]$ be a nonzero probability.

(1) $(\sigma_1,\gamma,\sigma_1',p) \in T$ and $\gamma \mid I \neq \langle\rangle$ and $\gamma \notin v \Rightarrow \sigma_1 \approx \sigma_1'$

(2) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow P_{\langle v,\approx\rangle}(\sigma_1,\gamma,\sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2,\gamma,\sigma_1')$.

We use this state machine formalization and definition of P-restrictiveness throughout the remainder of this report.

McCullough's state machine formalization and restrictiveness can be similarly generalized to allow transitions on atomic sequences of events as follows.

Definition: A *state machine* Σ is given by a six tuple $(S, \sigma_0, E, I, O, T)$, where S is the set of all possible states, $\sigma_0 \in S$ is the initial state, E is the set of possible events, $I \subseteq E$ is the set of all input events, $O \subseteq E$ is the set of all output events, and $T \subseteq S \times E^* \times S$ is the set of all possible state transitions.

Definition: Extended transitions are given by $ET \subseteq S \times E^* \times S$ where $(\sigma_1, \langle e_1, \ldots, e_n \rangle, \sigma_n) \in ET$ if and only if there exists some sequence of states $\sigma_2, \ldots, \sigma_{n-1}$, such that $(\sigma_i, e_i, \sigma_{i+1}) \in T$ for all i, $0 < i < n-1$.

Definition: Let $\approx$ be an equivalence relation on states of a system Σ and v be a subset of E^*. The projection $\langle v, \approx \rangle$ is *restrictive* for Σ if the following condition holds.

Let $(\sigma_1, \gamma, \sigma_1')$ be an arbitrary transition of Σ.

(1) $\gamma \ominus I$ and $\gamma \notin v \Rightarrow \sigma_1 \approx \sigma_1'$ and

(2) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow (\exists \sigma_2' \in S)(\exists \gamma' \in E^*)$

$[$(2a) $(\sigma_2, \gamma', \sigma_2') \in ET$,
(2b) $\sigma_2' \approx \sigma_1'$,
(2c) $\gamma \ominus I$ and $\gamma \in v \Rightarrow \gamma' = \gamma$,
(2d) $\gamma \notin v \Rightarrow \gamma' \in (E^* - v)^*$, and
(2e) $\gamma \not\ominus I$ and $\gamma \in v \Rightarrow (\exists \gamma_1, \gamma_2 \in ((E - I)^* - v)^*)[\gamma' = \gamma_1 \hat{} \gamma \hat{} \gamma_2]]$.

5.3 Existing Solutions

Solutions for the secure readers-writers problem that use event counts have appeared in the literature since 1974 [10-12], and [8]. These solutions allow the writer to start writing at any time, regardless of whether a reader is currently reading. This prevents all interference with low outputs by high inputs. However, it has the unfortunate consequence that writers can deny service to readers by frequent writing.

The following solution is equivalent in effect to these event count solutions.

Let $\Sigma 2$ be the state machine given by $(S, \sigma_0, E, I, O, T)$, where

$$S = \{0,1\} \times \{0,1\} \times \textbf{object} \times \textbf{integer} \times \textbf{integer}$$

The state of this system is made up of two Booleans, one object (we assume that the type **object** is previously defined) and two integers. To make the system easier to describe and to understand, we refer to the components of a state σ by the following mnemonics:

$\sigma.LoLock$: **boolean**
$\sigma.HiWaiting$: **boolean**
$\sigma.O$: **object**
$\sigma.EventCount$: **integer**
$\sigma.HiStartRead$: **integer**

The initial state of the system is given by:

$\sigma_0.LoLock =$ **false** {Note: **false** means 0, **true** means 1}
$\sigma_0.HiWaiting =$ **false**
$\sigma_0.O =$ **null**
$\sigma_0.EventCount = 0$
$\sigma_0.HiStartRead = 0$

$E = \{BeginRead, OKtoRead, Read, EndRead, ReadSuccessful, ReadFailed, BeginWrite, OKtoWrite, ObjectWritten, ObjectNotWritten, EndWrite, WriteSuccessful, \epsilon\} \cup$ **object** $\cup$ $\{ write\ o \mid o \in$ **object** $\}$

$I = \{BeginRead, Read, EndRead, BeginWrite, EndWrite\} \cup \{ Write\, o \mid o \in \textbf{object} \}$

$O = \{OKtoRead, ReadSuccessful, ReadFailed, OKtoWrite, ObjectWritten, ObjectnotWritten, WriteSuccessful\} \cup \textbf{object}$

$T = \{ (\sigma, \langle BeginRead \rangle, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \} \cup$

$\{ (\sigma, \langle OKtoRead \rangle, \sigma', .143) \mid \sigma.HiWaiting = \textbf{true} \text{ and } \sigma.LoLock = \textbf{false} \text{ and } \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{false} \text{ and } \sigma'.HiStartRead = \sigma.EventCount \} \cup$
$\{ (\sigma, \langle \epsilon \rangle, \sigma, .143) \mid (\sigma.HiWaiting = \textbf{false} \text{ or } \sigma.LoLock = \textbf{true}) \} \cup$
$\{ (\sigma, \langle Read, o \rangle, \sigma, .143) \mid o = \sigma.O \} \cup$
$\{ (\sigma, \langle EndRead, ReadSuccessful \rangle, \sigma, .143) \mid \sigma.HiStartRead = \sigma.EventCount \} \cup$
$\{ (\sigma, \langle EndRead, ReadFailed \rangle, \sigma, .143) \mid \sigma.HiStartRead \neq \sigma.EventCount \} \cup$

$\{ (\sigma, \langle BeginWrite, OKtoWrite \rangle, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{true} \text{ and } \sigma'.EventCount = \sigma.EventCount + 1 \} \cup$

$\{ (\sigma, \langle Write\, o, ObjectWritten \rangle, \sigma', .143) \mid \sigma.LoLock = \textbf{true} \text{ and } o \in \textbf{object} \text{ and } \sigma' = \sigma \text{ except } \sigma'.O = o \} \cup$
$\{ (\sigma, \langle Write\, o, ObjectNotWritten \rangle, \sigma, .143) \mid \sigma.LoLock = \textbf{false} \} \cup$

$\{ (\sigma, \langle EndWrite, WriteSuccessful \rangle, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{false} \}.$

Note: the set $\{ (\sigma, \langle \epsilon \rangle, \sigma, .143) \mid (\sigma.HiWaiting = \textbf{false} \text{ or } \sigma.LoLock = \textbf{true}) \}$ is included in T so that $\Sigma 2$ will be P-restrictive.

Theorem 4: Let $\Sigma 2' = (S', \sigma_0', E', I', O', T')$ where $S' = S$, $\sigma_0' = \sigma_0$, $E' = E$, $I' = I$, $O' = O$, and $T' = \{ (\sigma_1, \gamma, \sigma_2) \mid \exists p \in (0, 1] \text{ such that } (\sigma_1, \gamma, \sigma_2, p) \in T \}$.

Let $\approx$ be defined by:

For all σ and σ', $\sigma \approx \sigma'$ if and only if
$\sigma.LoLock = \sigma'.LoLock$,

and let

$v = \{\langle BeginWrite, OKtoWrite \rangle, \langle EndWrite, WriteSuccessful \rangle\} \cup \{ \langle Write\, o, ObjectWritten \rangle \mid o \in \textbf{object} \} \cup \{ \langle Write\, o, ObjectNotWritten \rangle \mid o \in \textbf{object} \}.$

The projection $\langle v, \approx \rangle$ is restrictive for $\Sigma 2'$.

Proof: Let $(\sigma_1, \gamma, \sigma_1')$ be an arbitrary transition of $\Sigma 2'$.

We must show that:

(1) $\gamma \Leftrightarrow I'$ and $\gamma \notin v \Rightarrow \sigma_1 \approx \sigma_1'$ and

(2) $\forall \sigma_2 \in S', \sigma_1 \approx \sigma_2 \Rightarrow (\exists \sigma_2' \in S')(\exists \gamma' \in E'^*)$

[(2a) $(\sigma_2, \gamma', \sigma_2') \in ET'$,
(2b) $\sigma_2' \approx \sigma_1'$,
(2c) $\gamma \Leftrightarrow I'$ and $\gamma \in v \Rightarrow \gamma' = \gamma$,

(2d) $\gamma \notin v \Rightarrow \gamma' \in (E'^* - v)^*$, and

(2e) $\gamma \not\triangleleft\!\triangleright I'$ and $\gamma \in v \Rightarrow (\exists \gamma_1, \gamma_2 \in (E'^* - v)^*)[\gamma' = \gamma_1{}^\wedge\gamma^\wedge\gamma_2]]$.

To show (1), we examine the definition of T' to find all γ such that $(\sigma_1, \gamma, \sigma_1') \in T'$ and $\gamma \in \triangleright I$ and $\gamma \notin v$. The examination reveals that there are four such γ : $\langle BeginRead \rangle$, $\langle Read, o \rangle$, $\langle EndRead, ReadSuccessful \rangle$, and $\langle EndRead, ReadFailed \rangle$. We consider the four cases individually.

Case 1: $\gamma = \langle BeginRead \rangle$.
The only state transitions that accept *BeginRead* as input are given by:

$$\{ (\sigma, \langle BeginRead \rangle, \sigma') \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \}$$

Thus, $\sigma_1' = \sigma_1$ except $\sigma_1'.HiWaiting = \textbf{true}$. And by the definition of $\approx$, $\sigma_1 \approx \sigma_1'$.

Case 2: $x = \langle Read, o \rangle$.
The only state transitions that engage in $\langle Read, o \rangle$ are given by:

$$\{ (\sigma, \langle Read, o \rangle, \sigma) \mid o = \sigma.O \}$$

Thus there is no change in state, and so, $\sigma_1 \approx \sigma_1'$.

Case 3: $x = \langle EndRead, ReadSuccessful \rangle$.
The only state transitions that engage in $\langle EndRead, ReadSuccessful \rangle$ are given by:

$$\{ (\sigma, \langle EndRead, ReadSuccessful \rangle, \sigma) \mid \sigma.HiStartRead = \sigma.EventCount \}$$

Thus there is no change in state and so, $\sigma_1 \approx \sigma_1'$.

Case 4: $x = \langle EndRead, ReadFailed \rangle$.
The only state transitions that engage in $\langle EndRead, ReadFailed \rangle$ are given by:

$$\{ (\sigma, \langle EndRead, ReadFailed \rangle, \sigma) \mid \sigma.HiStartRead \neq \sigma.EventCount \}$$

. Thus there is no change in state and so, $\sigma_1 \approx \sigma_1'$.

Therefore, (1) holds.

Now, to show (2), let σ_2 be an arbitrary state such that $\sigma_1 \approx \sigma_2$. We must show that

$(\exists \sigma_2' \in S')(\exists \gamma' \in E'^*)$

$[$(2a) $(\sigma_2, \gamma', \sigma_2') \in ET'$,
(2b) $\sigma_2' \approx \sigma_1'$,
(2c) $\gamma \in\!\triangleright I'$ and $\gamma \in v \Rightarrow \gamma' = \gamma$,
(2d) $\gamma \notin v \Rightarrow \gamma' \in (E'^* - v)^*$, and
(2e) $\gamma \not\in\!\triangleright I'$ and $\gamma \in v \Rightarrow (\exists \gamma_1, \gamma_2 \in (E'^* - v)^*)[\gamma' = \gamma_1{}^\wedge\gamma^\wedge\gamma_2]]$.

By examination of T', the transitions of $\Sigma 2'$ are described by ten sets of transitions unioned together. By showing (2) for all 10 sets we will have shown (2) for all transitions. We consider the 10 sets in 10 separate cases.

Case 1: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle BeginRead \rangle, \sigma') \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \}$

Choose $\sigma_2' = \sigma_2$ **except** $\sigma 2'.HiWaiting = \textbf{true}$. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle BeginRead \rangle, \sigma') \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \ominus I'$ and $\gamma \in v$. Therefore, Case 1 holds.

Case 2: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle OKtoRead \rangle, \sigma') \mid \sigma.HiWaiting = \textbf{true}$ and $\sigma.LoLock = \textbf{false}$ and $\sigma' = \sigma$ except $\sigma'.HiWaiting = \textbf{false}$ and $\sigma'.HiStartRead = \sigma.EventCount \}$

Suppose that $\sigma_2.HiWaiting = \textbf{true}$. Then, choose $\sigma_2' = \sigma_2$ **except** $\sigma_2'.HiWaiting = \textbf{false}$ and $\sigma_2'.HiStartRead = \sigma_2.EventCount$. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle OKtoRead \rangle, \sigma') \mid \sigma.HiWaiting = \textbf{true}$ and $\sigma.LoLock = \textbf{false}$ and $\sigma' = \sigma$ except $\sigma'.HiWaiting = \textbf{false}$ and $\sigma'.HiStartRead = \sigma.EventCount \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \ominus I'$ and $\gamma \in v$.
On the other hand, suppose that $\sigma_2.HiWaiting = \textbf{false}$. Then, choose $\sigma_2' = \sigma_2$ and choose $\gamma' = \langle \epsilon \rangle$. Now, $(\sigma_2, \gamma', \sigma_2') \in \{ (\sigma, \langle \epsilon \rangle, \sigma) \mid (\sigma.HiWaiting = \textbf{false}$ or $\sigma.LoLock = \textbf{true}) \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; (2c) and (2e) hold vacuously since $\gamma \notin v$; and (2d) holds since $\gamma' \in (E'^* - v)^*$. Therefore, Case 2 holds.

Case 3: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle \epsilon \rangle, \sigma) \mid (\sigma.HiWaiting = \textbf{false}$ or $\sigma.LoLock = \textbf{true}) \}$

This case is analogous to Case 2.

Case 4: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle Read, o \rangle, \sigma) \mid o = \sigma.O \}$

Choose $\sigma_2' = \sigma_2$. Choose $\gamma' = \langle Read, o' \rangle$ where $o' = \sigma_2.O$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle BeginRead \rangle, \sigma') \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \}$
so (2a) holds; by the reflexivity and the transitivity of $\approx$, (2b) holds; (2c) and (2e) hold vacuously since $\gamma \notin v$; and (2d) holds since $\gamma' \in (E'^* - v)^*$. Therefore, Case 4 holds.

Case 5: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle EndRead, ReadSuccessful \rangle, \sigma) \mid \sigma.HiStartRead = \sigma.EventCount \}$

Suppose that $\sigma_2.HiStartRead = \sigma_2.EventCount$. Then, choose $\sigma_2' = \sigma_2$. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in \{ (\sigma, \langle EndRead, ReadSuccessful \rangle, \sigma) \mid \sigma.HiStartRead = \sigma.EventCount \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; (2c) and (2e) hold vacuously since $\gamma \notin v$; and (2d) holds since $\gamma' \in (E'^* - v)^*$.
On the other hand, suppose that $\sigma_2.HiStartRead \neq \sigma_2.EventCount$. Then, choose $\sigma_2' = \sigma_2$ and choose $\gamma' = \langle EndRead, ReadFailed \rangle$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle EndRead, ReadFailed \rangle, \sigma) \mid \sigma.HiStartRead \neq \sigma.EventCount \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; (2c) and (2e) hold vacuously since $\gamma \notin v$; and (2d) holds since $\gamma' \in (E'^* - v)^*$. Therefore, Case 5 holds.

Case 6: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle EndRead, ReadFailed \rangle, \sigma) \mid \sigma.HiStartRead \neq \sigma.EventCount \}$

This case is analogous to Case 5.

Case 7: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle BeginWrite, OKtoWrite \rangle, \sigma') \mid \sigma' = \sigma$ except $\sigma'.LoLock =$ **true** and $\sigma'.EventCount = \sigma.EventCount + 1 \}$

Choose $\sigma_2' = \sigma_2$ **except** $\sigma 2'.LoLock =$ **true** and $\sigma 2'.EventCount = \sigma 2.EventCount + 1$. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in \{ (\sigma, \langle BeginWrite, OKtoWrite \rangle, \sigma') \mid \sigma' = \sigma$ except $\sigma'.LoLock =$ **true** and $\sigma'.EventCount = \sigma.EventCount + 1 \}$
so (2a) holds; since $\sigma_1'.LoLock =$ **true** $= \sigma_2'.LoLock$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \Leftrightarrow I'$ and $\gamma \in v$. Therefore, Case 7 holds.

Case 8: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle Write\ o, ObjectWritten \rangle, \sigma') \mid \sigma.LoLock =$ **true** and $o \in$ **object** and $\sigma' = \sigma$ except $\sigma'.O = o \}$

Choose $\sigma_2' = \sigma_2$ **except** $\sigma 2'.O = o$. Choose $\gamma' = \gamma$. Now, since $\sigma_2 \approx \sigma_1$, $\sigma_2.LoLock =$ **true** and $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle Write\ o, ObjectWritten \rangle, \sigma') \mid \sigma.LoLock =$ **true** and $o \in$ **object** and $\sigma' = \sigma$ except $\sigma'.O = o \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \Leftrightarrow I'$ and $\gamma \in v$. Therefore, Case 8 holds.

Case 9: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle Write\ o, ObjectNotWritten \rangle, \sigma) \mid \sigma.LoLock =$ **false** $\}$

Choose $\sigma_2' = \sigma_2$. Choose $\gamma' = \gamma$. Now, since $\sigma_2 \approx \sigma_1$, $\sigma_2.LoLock =$ **false** and $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle Write\ o, ObjectNotWritten \rangle, \sigma) \mid \sigma.LoLock =$ **false** $\}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \Leftrightarrow I'$ and $\gamma \in v$. Therefore, Case 9 holds.

Case 10: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle EndWrite, WriteSuccessful \rangle, \sigma') \mid \sigma' = \sigma$ except $\sigma'.LoLock =$ **false** $\}$

Choose $\sigma_2' = \sigma_2$ **except** $\sigma 2'.LoLock =$ **false**. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle EndWrite, WriteSuccessful \rangle, \sigma') \mid \sigma' = \sigma$ except $\sigma'.LoLock =$ **false** $\}$
so (2a) holds; since $\sigma_1'.LoLock =$ **false** $= \sigma_2'$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \Leftrightarrow I'$ and $\gamma \in v$. Therefore, Case 10 holds.

Thus (2) holds and $\langle v, \approx \rangle$ is restrictive for $\Sigma 2'$. []

Theorem 5: Let $\approx$ be defined by:

For all σ and σ', $\sigma \approx \sigma'$ if and only if $\sigma.LoLock = \sigma'.LoLock$

and let
$v = \{ \langle BeginWrite, OKtoWrite \rangle, \langle EndWrite, WriteSuccessful \rangle \} \cup \{ \langle Write\ o, ObjectWritten \rangle \mid o \in$ **object** $\} \cup \{ \langle Write\ o, ObjectNotWritten \rangle \mid o \in$ **object** $\}$

The projection $\langle v, \approx \rangle$ is P-restrictive for $\Sigma 2$.

Proof: Let σ_1 and $\sigma_1' \in S$ be arbitrary states, $\gamma \in E^*$ be an arbitrary event sequence, and $p \in (0, 1]$ be a nonzero probability.

We must show that:

(1) $(\sigma_1, \gamma, \sigma_1', p) \in T$ and $\gamma \mid I \neq \langle\rangle$ and $\gamma \notin v \Rightarrow \sigma_1 \approx \sigma_1'$ and
(2) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow$

$$P_{\langle v, \approx \rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v, \approx \rangle}(\sigma_2, \gamma, \sigma_1').$$

To show (1), we examine the definition of T to find all γ such that $(\sigma_1, \gamma, \sigma_1', p) \in T$ and $\gamma \mid I \neq \langle\rangle$ and $\gamma \notin v$. The examination reveals that there are four such γ : $\langle BeginRead\rangle$, $\langle Read, o\rangle$, $\langle EndRead, ReadSuccessful\rangle$, and $\langle EndRead, ReadFailed\rangle$. We consider the four cases individually.

Case 1: $\gamma = \langle BeginRead\rangle$.
The only state transitions that accept $BeginRead$ as input are given by:

$$\{\, (\sigma, \langle BeginRead\rangle, \sigma', .143) \mid \sigma' = \sigma \text{ except} \sigma'.HiWaiting = \textbf{true} \,\}$$

Thus, $\sigma_1' = \sigma_1$ except $\sigma_1'.HiWaiting = \textbf{true}$. And by the definition of $\approx$, $\sigma_1 \approx \sigma_1'$.

Case 2: $x = \langle Read, o\rangle$.
The only state transitions that engage in $\langle Read, o\rangle$ are given by:

$$\{\, (\sigma, \langle Read, o\rangle, \sigma, .143) \mid o = \sigma.O \,\}.$$

Thus there is no change in state, and so, $\sigma_1 \approx \sigma_1'$.

Case 3: $x = \langle EndRead, ReadSuccessful\rangle$.
The only state transitions that engage in $\langle EndRead, ReadSuccessful\rangle$ are given by

$$\{\, (\sigma, \langle EndRead, ReadSuccessful\rangle, \sigma, .143) \mid \sigma.HiStartRead = \sigma.EventCount \,\}.$$

Thus there is no change in state and so, $\sigma_1 \approx \sigma_1'$.

Case 4: $x = \langle EndRead, ReadFailed\rangle$.
The only state transitions that engage in $\langle EndRead, ReadFailed\rangle$ are given by

$$\{\, (\sigma, \langle EndRead, ReadFailed\rangle, \sigma, .143) \mid \sigma.HiStartRead \neq \sigma.EventCount \,\}.$$

Thus there is no change in state and so, $\sigma_1 \approx \sigma_1'$.

Therefore, (1) holds.

Now, to show (2), let σ_2 be an arbitrary state such that $\sigma_1 \approx \sigma_2$. We must show that $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

We have two major cases: $\gamma \in v$ and $\gamma \notin v$.

Case 1: $\gamma \in v$.
According to the definition of v, there are four different event sequences $\gamma \in v$ for which we must show the above equality. We proceed with one subcase for each of these event sequences.

Case 1.1: $\gamma = \langle BeginWrite, OKtoWrite\rangle$.
By examination of T, the transitions that can engage in γ are given by:

$$\{\, (\sigma, \gamma, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{true} \text{ and } \sigma'.EventCount = \sigma.EventCount + 1 \,\}$$

Suppose $\sigma_1'.LoLock = \textbf{true}$. There exists exactly one $\sigma' \in S$ such that $\sigma' = \sigma_1$ except $\sigma'.LoLock = \textbf{true}$ and $\sigma'.EventCount = \sigma_1.EventCount + 1$. Since $\sigma_1'.LoLock = \textbf{true} = \sigma'.LoLock$, $\sigma_1' \approx \sigma'$ and therefore, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .143$. By similar reasoning, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .143$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

Suppose, on the other hand, that $\sigma_1'.LoLock = \textbf{false}$. In this case, there does not exist a $\sigma' \approx \sigma_1'$ such that $\sigma' = \sigma_1$ except $\sigma'.LoLock = \textbf{true}$ and $\sigma'.EventCount = \sigma_1.EventCount+1$. And so, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. And by similar reasoning, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence again, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$. Therefore, Case 1.1 holds.

Case 1.2: $\gamma = \langle Write\, o, ObjectWritten\rangle$ for some object o.
By examination of T, the transitions that can engage in γ are given by

$$\{\, (\sigma, \gamma, \sigma', .143) \mid \sigma.LoLock = \textbf{true} \;\text{ and }\; \sigma' = \sigma \text{ except } \sigma'.O = o \,\}.$$

Suppose that $\sigma_1.LoLock = \sigma_1'.LoLock = \textbf{true}$. There is exactly one state σ' such that $\sigma' = \sigma_1'$ except $\sigma'.O = o$. Since $(\sigma_1, \langle Write\, o, ObjectWritten\rangle, \sigma', .143)$ is thus a member of the above set and $\sigma' \approx \sigma_1'$, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .143$. By the same reasoning (since $\sigma_1 \approx \sigma_2$ and hence, $\sigma_2.LoLock = \sigma_1'.LoLock = \textbf{true}$ also), $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .143$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

On the other hand, suppose that $\sigma_1.LoLock = \textbf{false}$ or $\sigma_1'.LoLock = \textbf{false}$. In this case, there does not exist a $\sigma' \approx \sigma_1'$ such that $\sigma_1.LoLock = \textbf{true}$ and $\sigma' = \sigma_1$ except $\sigma'.O = o$, and so, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. Similarly, since $\sigma_1 \approx \sigma_2$ and so $\sigma_2.LoLock = \sigma_1.LoLock$, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence, again $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$. Therefore, Case 1.2 holds.

Case 1.3: $\gamma = \langle Write\, o, ObjectNotWritten\rangle$ for some object o.
By examination of T, the transitions that can engage in γ are given by:

$$\{\, (\sigma, \gamma, \sigma, .143) \mid \neg\sigma.LoLock \,\}.$$

Suppose that $\sigma_1.LoLock = \sigma_1'.LoLock = \textbf{false}$. Then, $(\sigma_1, \gamma, \sigma_1, .143)$ is a member of the above set and $\sigma_1 \approx \sigma_1'$, and so $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .143$. By the same reasoning (since $\sigma_1 \approx \sigma_2$ and hence, $\sigma_2.LoLock = \sigma_1'.LoLock = \textbf{false}$ also), $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .143$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

On the other hand, suppose that $\sigma_1.LoLock = \textbf{true}$ or $\sigma_1'.LoLock = \textbf{true}$. In this case, either $(\sigma_1, \gamma, \sigma_1, .143)$ is not a member of the above set, or $\sigma_1 \not\approx \sigma_1'$, and so, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. Similarly, since $\sigma_1 \approx \sigma_2$ and so $\sigma_2.LoLock = \sigma_1.LoLock$, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence again, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$. Therefore, Case 1.3 holds.

Case 1.4: $\gamma = \langle EndWrite, WriteSuccessful\rangle$.
By examination of T, the transitions that can engage in γ are given by

$$\{\, (\sigma, \gamma, \sigma', .143) \mid \sigma' = \sigma \text{ except} \sigma'.LoLock = \textbf{false} \,\}.$$

Suppose $\sigma_1'.LoLock = \textbf{false}$. Then, there is exactly one state σ' such that $\sigma' = \sigma_1$ except $\sigma'.LoLock = \textbf{false}$. Since $\sigma_1'.LoLock = \textbf{false} = \sigma'.LoLock$, $\sigma_1' \approx \sigma'$ and therefore, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .143$. By similar reasoning, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .143$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

Suppose, on the other hand, that $\sigma_1'.LoLock = \textbf{true}$. In this case, there does not exist a $\sigma' \approx \sigma_1'$ such that $\sigma' = \sigma_1$ except $\sigma'.LoLock = \textbf{false}$. And so, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. And by similar reasoning, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence again, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$. Therefore, Case 1.4 holds, and so Case 1 holds.

Case 2: $\gamma \notin v$.
We divide this case into two subcases: $\sigma_1 \approx \sigma_1'$ and $\sigma_1 \not\approx \sigma_1'$.

Case 2.1: $\sigma_1 \approx \sigma_1'$
By the definitions of T, v, and $\approx$, it can be shown that for any possible transition $(\sigma, \gamma, \sigma', p)$ where γ is an invisible event sequence, it is the case that $\sigma \approx \sigma'$ (i.e., for any $\gamma' \in E^* - v$, $(\sigma_1, \gamma', \sigma_2', p) \in T$ implies $\sigma_1 \approx \sigma_2'$).
Now, by the definition of P,

$$P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = \sum_{\substack{\gamma' \in E^* - v \text{ and} \\ \sigma_2' \approx \sigma_1'}} p_{(\sigma_1, \gamma', \sigma_2')}$$

Since $\sigma_1 \approx \sigma_1'$ and, for any $\gamma' \in E^* - v$, $(\sigma_1, \gamma', \sigma_2', p) \in T$ implies $\sigma_1 \approx \sigma_2'$ (as noted above), the above equation can be simplified to

$$P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = \sum_{\gamma' \in E^* - v} p_{(\sigma_1, \gamma', \sigma_2')} = P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1)$$

Claim: Given that $\gamma \notin v$, for any $\sigma \in S$, $P_{\langle v,\approx\rangle}(\sigma, \gamma, \sigma) = .572$.

Justification: Given any state σ, (1) the event $\langle BeginRead \rangle$ can occur with probability .143; (2) The event $\langle Read, o \rangle$ can occur with probability .143; (3) Either $\langle OKtoRead \rangle$ or $\langle \epsilon \rangle$, but not both, can occur with probability .143 (depending on the values of $\sigma.HiWaiting$ and $\sigma.LoLock$); and (4) Either $\langle EndRead, ReadSuccessful \rangle$ or $\langle EndRead, ReadFailed \rangle$, but not both, can occur with probability .143 (depending on the values of $\sigma.HiStartRead$ and $\sigma.EventCount$).

Summing up these four, $P_{\langle v,\approx\rangle}(\sigma, \gamma, \sigma) = .572$, regardless of the state σ.

Therefore, we have,

$$P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1) = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_2) = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$$

and Case 2.1 holds.

Case 2.2: $\sigma_1 \not\approx \sigma_1'$
In this case, there is no $\sigma_2' \approx \sigma_1'$ and probability p, such that $(\sigma_1, \gamma, \sigma_2', p) \in T$. So, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. Similarly, since $\sigma_1 \approx \sigma_2$ and so, $\sigma_2 \not\approx \sigma_1'$, it can also be shown that $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Thus, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$ and Case 2.2 holds.

Thus (2) holds and $\langle v, \approx \rangle$ is P-restrictive for $\Sigma 2$. []

5.4 A First Attempt at Preventing Denial of Service

The above solution has no probabilistic interference. However, as mentioned previously, low writers can easily deny service to high readers by writing frequently. In fact, Reed and Kanodia [8] point out that "No algorithm can simultaneously guarantee that readers will be able to complete reading and that readers can never signal writers ..."

A reasonable approach to partially solving this denial of service problem is to nondeterministically decide whether to grant write access to the low writer. If the low writer were not always permitted to obtain write access, then the high reader would have a greater chance to complete reading.

A system designer might (maliciously or with good intentions) decide that if a high reader is currently reading, then the low writer should most often be denied write access. Whereas if the high reader is not reading, then the low reader should most often be granted access.

With this strategy in mind, the following solution might be advanced.

Let $\Sigma 3$ be the state machine given by $(S, \sigma_0, E, I, O, T)$, where

$$S = \{0,1\} \times \{0,1\} \times \{0,1\} \times \textbf{object} \times \textbf{integer} \times \textbf{integer}.$$

We refer to the components of a state σ by the following mnemonics:

$\sigma.LoLock$: **boolean**
$\sigma.HiWaiting$: **boolean**
$\sigma.HiReading$: **boolean**
$\sigma.O$: **object**
$\sigma.EventCount$: **integer**
$\sigma.HiStartRead$: **integer**.

The initial state of the system is given by

$\sigma_0.LoLock =$ **false**
$\sigma_0.HiWaiting =$ **false**
$\sigma_0.HiReading =$ **false**
$\sigma_0.O =$ **null**
$\sigma_0.EventCount = 0$
$\sigma_0.HiStartRead = 0$

$$E = \{BeginRead, OKtoRead, Read, EndRead, ReadSuccessful, ReadFailed, BeginWrite, OKtoWrite, NotOKtoWrite, ObjectWritten, ObjectNotWritten, EndWrite, WriteSuccessful, \epsilon\} \cup \textbf{object} \cup \{\, write\ o \mid o \in \textbf{object} \,\}$$

$$I = \{BeginRead, Read, EndRead, BeginWrite, EndWrite\} \cup \{\, Write\ o \mid o \in \textbf{object} \,\}$$

$$O = \{OKtoRead, ReadSuccessful, ReadFailed, OKtoWrite, NotOKtoWrite, ObjectWritten, ObjectnotWritten, WriteSuccessful\} \cup \textbf{object}$$

$$T = \{\, (\sigma, \langle BeginRead\rangle, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \,\} \cup$$
$$\{\, (\sigma, \langle OKtoRead\rangle, \sigma', .143) \mid \sigma.HiWaiting = \textbf{true} \text{ and } \sigma.LoLock = \textbf{false} \text{ and } \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{false} \text{ and } \sigma'.HiReading = \textbf{true} \text{ and } \sigma'.HiStartRead = \sigma.EventCount \,\} \cup$$
$$\{\, (\sigma, \langle \epsilon\rangle, \sigma, .143) \mid (\sigma.HiWaiting = \textbf{false} \text{ or } \sigma.LoLock = \textbf{true}) \,\} \cup$$
$$\{\, (\sigma, \langle Read, o\rangle, \sigma, .143) \mid o = \sigma.O \,\} \cup$$
$$\{\, (\sigma, \langle EndRead, ReadSuccessful\rangle, \sigma, .143) \mid \sigma.HiStartRead = \sigma.EventCount \,\} \cup$$
$$\{\, (\sigma, \langle EndRead, ReadFailed\rangle, \sigma, .143) \mid \sigma.HiStartRead \neq \sigma.EventCount \,\} \cup$$
$$\{\, (\sigma, \langle BeginWrite, OKtoWrite\rangle, \sigma', .043) \mid \sigma.HiReading = \textbf{true} \text{ and } \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{true} \text{ and } \sigma'.EventCount = \sigma.EventCount + 1 \,\} \cup$$

$\{ (\sigma, \langle BeginWrite, NotOKtoWrite \rangle, \sigma, .1) \mid \sigma.HiReading = \textbf{true} \} \cup$
$\{ (\sigma, \langle BeginWrite, OKtoWrite \rangle, \sigma', .1) \mid \sigma.HiReading = \textbf{false}$ and $\sigma' = \sigma$ except
$\sigma'.LoLock = \textbf{true}$ and $\sigma'.EventCount = \sigma.EventCount + 1 \} \cup$
$\{ (\sigma, \langle BeginWrite, NotOKtoWrite \rangle, \sigma, .043) \mid \sigma.HiReading = \textbf{false} \} \cup$
$\{ (\sigma, \langle Write\ o, ObjectWritten \rangle, \sigma', .143) \mid \sigma.LoLock = \textbf{true}$ and $\sigma' = \sigma$ except $\sigma'.O = o \} \cup$
$\{ (\sigma, \langle Write\ o, ObjectNotWritten \rangle, \sigma, .143) \mid \sigma.LoLock = \textbf{false} \} \cup$
$\{ (\sigma, \langle EndWrite, WriteSuccessful \rangle, \sigma', .143) \mid \sigma' = \sigma$ except $\sigma'.LoLock = \textbf{false} \}.$

Theorem 6: Let $\Sigma 3' = (S', \sigma_0', E', I', O', T')$ where $S' = S$, $\sigma_0' = \sigma_0$, $E' = E$, $I' = I$, $O' = O$, and $T' = \{ (\sigma_1, \gamma, \sigma_2) \mid \exists p \in (0, 1]$ such that $(\sigma_1, \gamma, \sigma_2, p) \in T \}$.

Let $\approx$ be defined by:

For all σ and σ', $\sigma \approx \sigma'$ if and only if $\sigma.LoLock = \sigma'.LoLock$

and let

$v = \{ \langle BeginWrite, OKtoWrite \rangle, \langle BeginWrite, NotOKtoWrite \rangle, \langle EndWrite, WriteSuccessful \rangle \} \cup$
$\{ \langle Write\ o, ObjectWritten \rangle \mid o \in \textbf{object} \} \cup \{ \langle Write\ o, ObjectNotWritten \rangle \mid o \in \textbf{object} \}.$

The projection $\langle v, \approx \rangle$ is restrictive for $\Sigma 3'$.

Proof: Let $(\sigma_1, \gamma, \sigma_1')$ be an arbitrary transition of $\Sigma 3'$.

We must show that:

(1) $\gamma \oplus\!\!\!\triangleright I'$ and $\gamma \notin v \Rightarrow \sigma_1 \approx \sigma_1'$ and

(2) $\forall \sigma_2 \in S', \sigma_1 \approx \sigma_2 \Rightarrow (\exists \sigma_2' \in S')(\exists \gamma' \in E'^*)$

$[$(2a) $(\sigma_2, \gamma', \sigma_2') \in ET'$,
(2b) $\sigma_2' \approx \sigma_1'$,
(2c) $\gamma \oplus\!\!\!\triangleright I'$ and $\gamma \in v \Rightarrow \gamma' = \gamma$,
(2d) $\gamma \notin v \Rightarrow \gamma' \in (E'^* - v)^*$, and
(2e) $\gamma \not\oplus\!\!\!\triangleright I'$ and $\gamma \in v \Rightarrow (\exists \gamma_1, \gamma_2 \in (E'^* - v)^*)[\gamma' = \gamma_1 \hat{} \gamma \hat{} \gamma_2]]$.

(1) can be shown in exactly the same way as in the proof of theorem 4.

Now, to show (2), let σ_2 be an arbitrary state such that $\sigma_1 \approx \sigma_2$. We must show that
$(\exists \sigma_2' \in S')(\exists \gamma' \in E'^*)$

$[$(2a) $(\sigma_2, \gamma', \sigma_2') \in ET'$,
(2b) $\sigma_2' \approx \sigma_1'$,
(2c) $\gamma \oplus\!\!\!\triangleright I'$ and $\gamma \in v \Rightarrow \gamma' = \gamma$,
(2d) $\gamma \notin v \Rightarrow \gamma' \in (E'^* - v)^*$, and
(2e) $\gamma \not\oplus\!\!\!\triangleright I'$ and $\gamma \in v \Rightarrow (\exists \gamma_1, \gamma_2 \in (E'^* - v)^*)[\gamma' = \gamma_1 \hat{} \gamma \hat{} \gamma_2]]$.

By examination of T', the transitions of $\Sigma 2'$ are described by 13 sets of transitions unioned together. By showing (2) for all 13 sets we will have shown (2) for all transitions. We consider the 13 sets in 13 separate cases.

Case 1: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle BeginRead \rangle, \sigma') \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \}$.

Choose $\sigma_2' = \sigma_2$ **except** $\sigma 2'.HiWaiting = \textbf{true}$. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle BeginRead \rangle, \sigma') \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \Leftarrow I'$ and $\gamma \in v$. Therefore, Case 1 holds.

Case 2: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle OKtoRead \rangle, \sigma') \mid \sigma.HiWaiting = \textbf{true}$ and $\sigma.LoLock = \textbf{false}$ and $\sigma' = \sigma$ except $\sigma'.HiWaiting = \textbf{false}$ and $\sigma'.HiReading = \textbf{true}$ and $\sigma'.HiStartRead = \sigma.EventCount \}$.

Suppose that $\sigma_2.HiWaiting = \textbf{true}$. Then, choose $\sigma_2' = \sigma_2$ **except** $\sigma_2'.HiWaiting = \textbf{false}$ and $\sigma_2'.HiReading = \textbf{true}$ and $\sigma_2'.HiStartRead = \sigma_2.EventCount$. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in \{ (\sigma, \langle OKtoRead \rangle, \sigma') \mid \sigma.HiWaiting = \textbf{true}$ and $\sigma.LoLock = \textbf{false}$ and $\sigma' = \sigma$ except $\sigma'.HiWaiting = \textbf{false}$ and $\sigma'.HiReading = \textbf{true}$ and $\sigma'.HiStartRead = \sigma.EventCount \}$ so (2a) holds; by the transitivity of $\approx$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \Leftarrow I'$ and $\gamma \in v$.

On the other hand, suppose that $\sigma_2.HiWaiting = \textbf{false}$. Then, choose $\sigma_2' = \sigma_2$ and choose $\gamma' = \langle \epsilon \rangle$. Now, $(\sigma_2, \gamma', \sigma_2') \in \{ (\sigma, \langle \epsilon \rangle, \sigma) \mid (\sigma.HiWaiting = \textbf{false}$ or $\sigma.LoLock = \textbf{true}) \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; (2c) and (2e) hold vacuously since $\gamma \notin v$; and (2d) holds since $\gamma' \in (E'^* - v)^*$. Therefore, Case 2 holds.

Case 3: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle \epsilon \rangle, \sigma) \mid (\sigma.HiWaiting = \textbf{false}$ or $\sigma.LoLock = \textbf{true}) \}$

This case is analagous to Case 2.

Case 4: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle Read, o \rangle, \sigma) \mid o = \sigma.O \}$

Choose $\sigma_2' = \sigma_2$. Choose $\gamma' = \langle Read, o' \rangle$ where $o' = \sigma_2.O$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle BeginRead \rangle, \sigma') \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \}$
so (2a) holds; by the reflexivity and the transitivity of $\approx$, (2b) holds; (2c) and (2e) hold vacuously since $\gamma \notin v$; and (2d) holds since $\gamma' \in (E'^* - v)^*$. Therefore, Case 4 holds.

Case 5: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle EndRead, ReadSuccessful \rangle, \sigma) \mid \sigma.HiStartRead = \sigma.EventCount \}$

Suppose that $\sigma_2.HiStartRead = \sigma_2.EventCount$. Then, choose $\sigma_2' = \sigma_2$. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle EndRead, ReadSuccessful \rangle, \sigma) \mid \sigma.HiStartRead = \sigma.EventCount \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; (2c) and (2e) hold vacuously since $\gamma \notin v$; and (2d) holds since $\gamma' \in (E'^* - v)^*$.
On the other hand, suppose that $\sigma_2.HiStartRead \neq \sigma_2.EventCount$. Then, choose $\sigma_2' = \sigma_2$ and choose $\gamma' = \langle EndRead, ReadFailed \rangle$. Now, $(\sigma_2, \gamma', \sigma_2') \in$
$\{ (\sigma, \langle EndRead, ReadFailed \rangle, \sigma) \mid \sigma.HiStartRead \neq \sigma.EventCount \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; (2c) and (2e) hold vacuously since $\gamma \notin v$; and (2d) holds since $\gamma' \in (E'^* - v)^*$. Therefore, Case 5 holds.

Case 6: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle EndRead, ReadFailed \rangle, \sigma) \mid \sigma.HiStartRead \neq \sigma.EventCount \}$
This case is analagous to Case 5.

Case 7: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle BeginWrite, OKtoWrite \rangle, \sigma') \mid \sigma.HiReading = \textbf{true}$ **and** $\sigma' = \sigma$ except $\sigma'.LoLock = \textbf{true}$ **and** $\sigma'.EventCount = \sigma.EventCount + 1 \}$

Choose $\sigma_2' = \sigma_2$ **except** $\sigma 2'.LoLock = \textbf{true}$ and $\sigma 2'.EventCount = \sigma 2.EventCount + 1$. Choose $\gamma' = \gamma$. Now, if $\sigma_2.HiReading = \textbf{true}$, then $(\sigma_2, \gamma', \sigma_2') \in$ $\{ (\sigma, \langle BeginWrite, OKtoWrite \rangle, \sigma') \mid \sigma.HiReading = \textbf{true}$ and $\sigma' = \sigma$ except $\sigma'.LoLock = \textbf{true}$ and $\sigma'.EventCount = \sigma.EventCount + 1 \}$. If $\sigma_2.HiReading = \textbf{false}$, then $(\sigma_2, \gamma', \sigma_2') \in \{ (\sigma, \langle BeginWrite, OKtoWrite \rangle, \sigma') \mid \sigma.HiReading = \textbf{false}$ and $\sigma' = \sigma$ except $\sigma'.LoLock = \textbf{true}$ and $\sigma'.EventCount = \sigma.EventCount + 1 \}$ so (2a) holds; since $\sigma_1'.LoLock = \textbf{true} = \sigma_2'.LoLock$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \not\in I'$ and $\gamma \in v$. Therefore, Case 7 holds.

Case 8: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle BeginWrite, NotOKtoWrite \rangle, \sigma) \mid \sigma.HiReading = \textbf{true} \}$
This case is analagous to Case 7.

Case 9: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle BeginWrite, OKtoWrite \rangle, \sigma') \mid \sigma.HiReading = \textbf{false}$ and $\sigma' = \sigma$ except $\sigma'.LoLock = \textbf{true}$ and $\sigma'.EventCount = \sigma.EventCount + 1 \}$
This case is analagous to Case 7.

Case 10: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle BeginWrite, NotOKtoWrite \rangle, \sigma) \mid \sigma.HiReading = \textbf{false} \}$
This case is analagous to case 7.

Case 11: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle Write\ o, ObjectWritten \rangle, \sigma') \mid \sigma.LoLock = \textbf{true}$ and $o \in$ **object** and $\sigma' = \sigma$ except $\sigma'.O = o \}$

Choose $\sigma_2' = \sigma_2$ **except** $\sigma 2'.O = o$. Choose $\gamma' = \gamma$. Now, since $\sigma_2 \approx \sigma_1$, $\sigma_2.LoLock = \textbf{true}$ and $(\sigma_2, \gamma', \sigma_2') \in \{ (\sigma, \langle Write\ o, ObjectWritten \rangle, \sigma') \mid \sigma.LoLock = \textbf{true}$ and $o \in$ **object** and $\sigma' = \sigma$ except $\sigma'.O = o \}$ so (2a) holds; by the transitivity of $\approx$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \not\in I'$ and $\gamma \in v$. Therefore, Case 11 holds.

Case 12: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle Write\ o, ObjectNotWritten \rangle, \sigma) \mid \sigma.LoLock = \textbf{false} \}$

Choose $\sigma_2' = \sigma_2$. Choose $\gamma' = \gamma$. Now, since $\sigma_2 \approx \sigma_1$, $\sigma_2.LoLock = \textbf{false}$ and $(\sigma_2, \gamma', \sigma_2') \in$ $\{ (\sigma, \langle Write\ o, ObjectNotWritten \rangle, \sigma) \mid \sigma.LoLock = \textbf{false} \}$
so (2a) holds; by the transitivity of $\approx$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \not\in I'$ and $\gamma \in v$. Therefore, Case 12 holds.

Case 13: $(\sigma_1, \gamma, \sigma_1') \in \{ (\sigma, \langle EndWrite, WriteSuccessful \rangle, \sigma') \mid \sigma' = \sigma$ except $\sigma'.LoLock = \textbf{false} \}$

Choose $\sigma_2' = \sigma_2$ **except** $\sigma 2'.LoLock = \textbf{false}$. Choose $\gamma' = \gamma$. Now, $(\sigma_2, \gamma', \sigma_2') \in$ $\{ (\sigma, \langle EndWrite, WriteSuccessful \rangle, \sigma') \mid \sigma' = \sigma$ except $\sigma'.LoLock = \textbf{false} \}$
so (2a) holds; since $\sigma_1'.LoLock = \textbf{false} = \sigma_2'$, (2b) holds; since $\gamma' = \gamma$, (2c) holds; and (2d) and (2e) hold vacuously since $\gamma \not\in I'$ and $\gamma \in v$. Therefore, Case 13 holds.

Thus (2) holds and $\langle v, \approx \rangle$ is restrictive for $\Sigma 2'$. []

Given the three objectives that the solution 1) be restrictive, 2) limit denial of service, and 3) provide good performance, $\Sigma 3$ is very reasonable. However, $\Sigma 3$ contains a probabilistic covert channel.

Theorem 7: Let $\approx$ and v be defined as in the previous theorem.

The projection $\langle v, \approx \rangle$ is not P-restrictive for $\Sigma 3$.

Proof: Let σ_1 be a state such that $\sigma_1.HiReading =$ **true**. Let σ_2 be a state such that $\sigma_2.HiReading =$ **false**. Additionally suppose that $\sigma_1 \approx \sigma_2$.

Let $\sigma_1' = \sigma_1$ except $\sigma_1'.LoLock =$ **true** and $\sigma_1'.EventCount = \sigma_1.EventCount + 1$. By the definitions of P and T,

$$P_{\langle v,\approx\rangle}(\sigma_1, \langle BeginWrite, OKtoWrite\rangle, \sigma_1') = .043$$

Let $\sigma_2' = \sigma_2$ except $\sigma_2'.LoLock =$ **true** and $\sigma_2'.EventCount = \sigma_2.EventCount + 1$. By the definitions of P and T,

$$P_{\langle v,\approx\rangle}(\sigma_2, \langle BeginWrite, OKtoWrite\rangle, \sigma_2') = .1$$

But since $\sigma_2' \approx \sigma_1'$,

$$P_{\langle v,\approx\rangle}(\sigma_2, \langle BeginWrite, OKtoWrite\rangle, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \langle BeginWrite, OKtoWrite\rangle, \sigma_1') = .1$$

If $\langle v, \approx \rangle$ were P-restrictive, then it would be the case that

$$P_{\langle v,\approx\rangle}(\sigma_1, \langle BeginWrite, OKtoWrite\rangle, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \langle BeginWrite, OKtoWrite\rangle, \sigma_1')$$

Since they are not equal, $\langle v, \approx \rangle$ is not P-restrictive for $\Sigma 3$. []

5.5 A P-Restrictive Solution

We now develop a solution to the secure readers-writers problem that limits denial of service and is P-restrictive.

Let $\Sigma 4$ be the state machine given by $(S, \sigma_0, E, I, O, T)$, where

$$S = \{0,1\} \times \{0,1\} \times \textbf{object} \times \textbf{integer} \times \textbf{integer}$$

We refer to the components of a state σ by the following mnemonics:

$\sigma.LoLock$: **boolean**
$\sigma.HiWaiting$: **boolean**
$\sigma.O$: **object**
$\sigma.EventCount$: **integer**
$\sigma.HiStartRead$: **integer**

The initial state of the system is given by:

$\sigma_0.LoLock =$ **false**
$\sigma_0.HiWaiting =$ **false**
$\sigma_0.O =$ **null**
$\sigma_0.EventCount = 0$
$\sigma_0.HiStartRead = 0$

$E = \{BeginRead, OKtoRead, Read, EndRead, ReadSuccessful, ReadFailed, BeginWrite, OKtoWrite, NotOKtoWrite, ObjectWritten, ObjectNotWritten, EndWrite, WriteSuccessful, \epsilon\} \cup$ **object** $\cup \{ write\ o \mid o \in$ **object** $\}$

$$I = \{BeginRead, Read, EndRead, BeginWrite, EndWrite\} \cup \{\, Write\ o \mid o \in \textbf{object} \,\}$$

$$O = \{OKtoRead, ReadSuccessful, ReadFailed, OKtoWrite, NotOKtoWrite, ObjectWritten, ObjectNotWritten, WriteSuccessful\} \cup \textbf{object}$$

$$\begin{aligned}
T = \ & \{\, (\sigma, \langle BeginRead\rangle, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \,\} \cup \\
& \{\, (\sigma, \langle OKtoRead\rangle, \sigma', .143) \mid \sigma.HiWaiting = \textbf{true} \text{ and } \sigma.LoLock = \textbf{false} \text{ and } \sigma' = \sigma \text{ except} \\
& \qquad \sigma'.HiWaiting = \textbf{false} \text{ and } \sigma'.HiStartRead = \sigma.EventCount \,\} \cup \\
& \{\, (\sigma, \langle\epsilon\rangle, \sigma, .143) \mid (\sigma.HiWaiting = \textbf{false} \text{ or } \sigma.LoLock = \textbf{true}) \,\} \cup \\
& \{\, (\sigma, \langle Read, o\rangle, \sigma, .143) \mid o = \sigma.O \,\} \cup \\
& \{\, (\sigma, \langle EndRead, ReadSuccessful\rangle, \sigma, .143) \mid \sigma.HiStartRead = \sigma.EventCount \,\} \cup \\
& \{\, (\sigma, \langle EndRead, ReadFailed\rangle, \sigma, .143) \mid \sigma.HiStartRead \neq \sigma.EventCount \,\} \cup \\
& \\
& \{\, (\sigma, \langle BeginWrite, OKtoWrite\rangle, \sigma', .71) \mid \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{true} \text{ and} \\
& \qquad \sigma'.EventCount = \sigma.EventCount + 1 \,\} \cup \\
& \{\, (\sigma, \langle BeginWrite, NotOKtoWrite\rangle, \sigma, .71) \mid \sigma \in S \,\} \cup \\
& \\
& \{\, (\sigma, \langle Write\ o, ObjectWritten\rangle, \sigma', .143) \mid \sigma.LoLock = \textbf{true} \text{ and } \sigma' = \sigma \text{ except } \sigma'.O = o \,\} \cup \\
& \{\, (\sigma, \langle Write\ o, ObjectNotWritten\rangle, \sigma, .143) \mid \sigma.LoLock = \textbf{false} \,\} \cup \\
& \\
& \{\, (\sigma, \langle EndWrite, WriteSuccessful\rangle, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{false} \,\}.
\end{aligned}$$

$\Sigma 4$ limits denial of service assuming that the low writer releases its write lock (i.e., performs an EndWrite) within some reasonable amount of time after obtaining it. If we cannot make this assumption (i.e., if the low writer is possibly erroneous or possibly malicious), then the probability of one of the existing transitions can be reduced by .01, and the following set can be added to T:

$$\{\, (\sigma, LockBroken, \sigma', .01) \mid \sigma.LoLock = \textbf{true} \text{ and } \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{false} \,\}.$$

With this additional transition, the system may at any time break the low writer's lock on the object, thus preventing the low writer from obtaining a lock on the object and never releasing it.

$\Sigma 4$ (with or without the additional set of transitions) contains no probabilistic interference.

Theorem 8: Let $\approx$ be defined by:

For all σ and σ', $\sigma \approx \sigma'$ if and only if
$\sigma.LoLock = \sigma'.LoLock$

and let
$v = \{\langle BeginWrite, OKtoWrite\rangle, \langle BeginWrite, NotOKtoWrite\rangle, \langle EndWrite, WriteSuccessful\rangle\} \cup \{\, \langle Write\ o, ObjectWritten\rangle \mid o \in \textbf{object} \,\} \cup \{\, \langle Write\ o, ObjectNotWritten\rangle \mid o \in \textbf{object} \,\}$

The projection $\langle v, \approx\rangle$ is P-restrictive for $\Sigma 4$.

Proof: Let σ_1 and $\sigma_1' \in S$ be arbitrary states, $\gamma \in E^*$ be an arbitrary event sequence, and $p \in (0, 1]$ be a nonzero probability.

We must show that:

(1) $(\sigma_1, \gamma, \sigma_1', p) \in T$ and $\gamma \Leftarrow I$ and $\gamma \notin v \Rightarrow \sigma_1 \approx \sigma_1'$, and

(2) $\forall \sigma_2 \in S, \sigma_1 \approx \sigma_2 \Rightarrow P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

To show (1), we examine the definition of T to find all γ such that $(\sigma_1, \gamma, \sigma_1', p) \in T$ and $\gamma \in\!\triangleright I$ and $\gamma \notin v$. The examination reveals that there are four such γ : $\langle BeginRead\rangle$, $\langle Read, o\rangle$, $\langle EndRead, ReadSuccessful\rangle$, and $\langle EndRead, ReadFailed\rangle$. We consider the four cases individually.

Case 1: $\gamma = \langle BeginRead\rangle$.
The only state transitions that accept $BeginRead$ as input are given by:

$$\{\, (\sigma, \langle BeginRead\rangle, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.HiWaiting = \textbf{true} \,\}$$

Thus, $\sigma_1' = \sigma_1$ except $\sigma_1'.HiWaiting = \textbf{true}$. And by the definition of $\approx$, $\sigma_1 \approx \sigma_1'$.

Case 2: $x = \langle Read, o\rangle$.
The only state transitions that engage in $\langle Read, o\rangle$ are given by:

$$\{\, (\sigma, \langle Read, o\rangle, \sigma, .143) \mid o = \sigma.O \,\}.$$

Thus there is no change in state, and so, $\sigma_1 \approx \sigma_1'$.

Case 3: $x = \langle EndRead, ReadSuccessful\rangle$.
The only state transitions that engage in $\langle EndRead, ReadSuccessful\rangle$ are given by:

$$\{\, (\sigma, \langle EndRead, ReadSuccessful\rangle, \sigma, .143) \mid \sigma.HiStartRead = \sigma.EventCount \,\}.$$

Thus there is no change in state and so, $\sigma_1 \approx \sigma_1'$.

Case 4: $x = \langle EndRead, ReadFailed\rangle$.
The only state transitions that engage in $\langle EndRead, ReadFailed\rangle$ are given by:

$$\{\, (\sigma, \langle EndRead, ReadFailed\rangle, \sigma, .143) \mid \sigma.HiStartRead \neq \sigma.EventCount \,\}.$$

Thus there is no change in state and so, $\sigma_1 \approx \sigma_1'$.

Therefore, (1) holds.

Now, to show (2), let σ_2 be an arbitrary state such that $\sigma_1 \approx \sigma_2$. We must show that $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

We have two major cases: $\gamma \in v$ and $\gamma \notin v$.

Case 1: $\gamma \in v$.
According to the definition of v, there are five different event sequences $\gamma \in v$ for which we must show the above equality. We proceed with one subcase for each of these event sequences.

Case 1.1: $\gamma = \langle BeginWrite, OKtoWrite\rangle$.
By examination of T, the transitions that can engage in γ are given by:
$\{\, (\sigma, \gamma, \sigma', .71) \mid \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{true} \text{ and } \sigma'.EventCount = \sigma.EventCount + 1 \,\}$

Suppose $\sigma_1'.LoLock = \textbf{true}$. There exists exactly one $\sigma' \in S$ such that $\sigma' = \sigma_1$ except $\sigma'.LoLock = \textbf{true}$ and $\sigma'.EventCount = \sigma_1.EventCount + 1$. Since $\sigma_1'.LoLock = \textbf{true} = \sigma'.LoLock$, $\sigma_1' \approx \sigma'$ and therefore, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .71$. By similar reasoning, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .71$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

Suppose, on the other hand, that $\sigma_1'.LoLock = \textbf{false}$. In this case, there does not exist a $\sigma' \approx \sigma_1'$ such that $\sigma' = \sigma_1$ except $\sigma'.LoLock = \textbf{true}$ and $\sigma'.EventCount = \sigma_1.EventCount + 1$. And so, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. And by similar reasoning, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence again, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$. Therefore, Case 1.1 holds.

Case 1.2: $\gamma = \langle BeginWrite, NotOKtoWrite\rangle$.
By examination of T, the transitions that can engage in γ are given by:

$$\{\, (\sigma, \gamma, \sigma, .71) \mid \sigma \in S \,\}.$$

Suppose $\sigma_1' \approx \sigma_1$. Then $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .71$. By the transitivity of $\approx$, $\sigma_1' \approx \sigma_2$ and so $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .71$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

Suppose, on the other hand, that $\sigma_1' \not\approx \sigma_1$. Then, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. By the transitivity of $\approx$, $\sigma_1' \not\approx \sigma_2$ and so $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence again, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

Therefore, Case 1.2 holds.

Case 1.3: $\gamma = \langle Write\ o, ObjectWritten\rangle$ for some object o.
By examination of T, the transitions that can engage in γ are given by:

$$\{\, (\sigma, \gamma, \sigma', .143) \mid \sigma.LoLock = \textbf{true} \quad \text{and } \sigma' = \sigma \text{ except } \sigma'.O = o \,\}.$$

Suppose that $\sigma_1.LoLock = \sigma_1'.LoLock = \textbf{true}$. There is exactly one state σ' such that $\sigma' = \sigma_1'$ except $\sigma'.O = o$. Since $(\sigma_1, \langle Write\ o, ObjectWritten\rangle, \sigma', .143)$ is thus a member of the above set and $\sigma' \approx \sigma_1'$, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .143$. By the same reasoning (since $\sigma_1 \approx \sigma_2$ and hence, $\sigma_2.LoLock = \sigma_1'.LoLock = \textbf{true}$ also), $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .143$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

On the other hand, suppose that $\sigma_1.LoLock = \textbf{false}$ or $\sigma_1'.LoLock = \textbf{false}$. In this case, there does not exist a $\sigma' \approx \sigma_1'$ such that $\sigma_1.LoLock = \textbf{true}$ and $\sigma' = \sigma_1$ except $\sigma'.O = o$, and so, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. Similarly, since $\sigma_1 \approx \sigma_2$ and so $\sigma_2.LoLock = \sigma_1.LoLock$, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence again, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$. Therefore, Case 1.3 holds.

Case 1.4: $\gamma = \langle Write\ o, ObjectNotWritten\rangle$ for some object o.
By examination of T, the transitions that can engage in γ are given by:

$$\{\, (\sigma, \gamma, \sigma, .143) \mid \sigma.LoLock = \textbf{false} \,\}.$$

Suppose that $\sigma_1.LoLock = \sigma_1'.LoLock = \textbf{false}$. Then, $(\sigma_1, \gamma, \sigma_1, .143)$ is a member of the above set, and $\sigma_1 \approx \sigma_1'$, and so $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .143$. By the same reasoning (since

$\sigma_1 \approx \sigma_2$ and hence, $\sigma_2.LoLock = \sigma_1'.LoLock =$ **false** also), $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .143$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

On the other hand, suppose that $\sigma_1.LoLock =$ **true** or $\sigma_1'.LoLock =$ **true**. In this case, either $(\sigma_1, \gamma, \sigma_1, .143)$ is not a member of the above set, or $\sigma_1 \not\approx \sigma_1'$, and so, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$.

Similarly, since $\sigma_1 \approx \sigma_2$ and so $\sigma_2.LoLock = \sigma_1.LoLock$, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence again, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$. Therefore, Case 1.4 holds.

Case 1.5: $\gamma = \langle EndWrite, WriteSuccessful\rangle$.
By examination of T, the transitions that can engage in γ are given by:

$$\{\, (\sigma, \gamma, \sigma', .143) \mid \sigma' = \sigma \text{ except } \sigma'.LoLock = \textbf{false} \,\}.$$

Suppose $\sigma_1'.LoLock =$ **false**. Then, there is exactly one state σ' such that $\sigma' = \sigma_1$ except $\sigma'.LoLock =$ **false**. Since $\sigma_1'.LoLock =$ **false** $= \sigma'.LoLock$, $\sigma_1' \approx \sigma'$ and therefore, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = .143$. By similar reasoning, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = .143$. Hence, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

Suppose, on the other hand, that $\sigma_1'.LoLock =$ **true**. In this case, there does not exist a $\sigma' \approx \sigma_1'$ such that $\sigma' = \sigma_1$ except $\sigma'.LoLock =$ **false**. And so, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = 0$. And by similar reasoning, $P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Hence again, $P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx\rangle}(\sigma_2, \gamma, \sigma_1')$.

Therefore, Case 1.5 holds, and so Case 1 holds.

Case 2: $\gamma \notin v$.
We will divide this case into two subcases: $\sigma_1 \approx \sigma_1'$ and $\sigma_1 \not\approx \sigma_1'$.

Case 2.1: $\sigma_1 \approx \sigma_1'$
By the definitions of T, v, and $\approx$, it can be shown that for any possible transition, $(\sigma, \gamma, \sigma', p)$ where γ is an invisible event sequence, it is the case that $\sigma \approx \sigma'$ (i.e., for any $\gamma' \in E^* - v$, $(\sigma_1, \gamma', \sigma_2', p) \in T$ implies $\sigma_1 \approx \sigma_2'$).

Now, by the definition of P,

$$P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = \sum_{\substack{\gamma' \in E^* - v \text{ and} \\ \sigma_2' \approx \sigma_1'}} p_{(\sigma_1, \gamma', \sigma_2')}$$

Since $\sigma_1 \approx \sigma_1'$ and, for any $\gamma' \in E^* - v$, $(\sigma_1, \gamma', \sigma_2', p) \in T$ implies $\sigma_1 \approx \sigma_2'$ (as noted above), the above equation can be simplified to:

$$P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1') = \sum_{\gamma' \in E^* - v} p_{(\sigma_1, \gamma', \sigma_2')} = P_{\langle v,\approx\rangle}(\sigma_1, \gamma, \sigma_1)$$

Claim: Given that $\gamma \notin v$, for any $\sigma \in S$, $P_{\langle v,\approx\rangle}(\sigma, \gamma, \sigma) = .572$.

Justification: Given any state σ, (1) the event $\langle BeginRead \rangle$ can occur with probability .143; (2) the event $\langle Read, o \rangle$ can occur with probability .143; (3) either $\langle OKtoRead \rangle$ or $\langle \epsilon \rangle$, but not both, can occur with probability .143 (depending on the values of $\sigma.HiWaiting$ and $\sigma.LoLock$). (4) either $\langle EndRead, ReadSuccessful \rangle$ or $\langle EndRead, ReadFailed \rangle$, but not both, can occur with probability .143 (depending on the values of $\sigma.HiStartRead$ and $\sigma.EventCount$). Summing up these four, $P_{\langle v,\approx \rangle}(\sigma, \gamma, \sigma) = .572$, regardless of the state σ.

Therefore, we have,

$$P_{\langle v,\approx \rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx \rangle}(\sigma_1, \gamma, \sigma_1) = P_{\langle v,\approx \rangle}(\sigma_2, \gamma, \sigma_2) = P_{\langle v,\approx \rangle}(\sigma_2, \gamma, \sigma_1')$$

and Case 2.1 holds.

Case 2.2: $\sigma_1 \not\approx \sigma_1'$

In this case, there does not exist a $\sigma_2' \approx \sigma_1'$ and a probability p, such that $(\sigma_1, \gamma, \sigma_2', p) \in T$. So, $P_{\langle v,\approx \rangle}(\sigma_1, \gamma, \sigma_1') = 0$. Similarly, since $\sigma_1 \approx \sigma_2$ and so, $\sigma_2 \not\approx \sigma_1'$, it can also be shown that $P_{\langle v,\approx \rangle}(\sigma_2, \gamma, \sigma_1') = 0$. Thus, $P_{\langle v,\approx \rangle}(\sigma_1, \gamma, \sigma_1') = P_{\langle v,\approx \rangle}(\sigma_2, \gamma, \sigma_1')$ and Case 2.2 holds.

Thus (2) holds and $\langle v, \approx \rangle$ is P-restrictive for $\Sigma 4$. []

6. Composing Systems

It is desirable for P-restrictiveness to be composable (as is restrictiveness). To show that P-restrictiveness is composable requires a formalization of the composition of probability-extended state machines. However, there is not only one way to define this composition. The main difficulty we encountered in defining the composition of machines was how to treat time. On the one hand timing considerations can affect the probabilities of events. For example, consider two systems: system A simply outputs a continuous sequence of 1's and system B simply outputs a continuous sequence of 0's. When these two systems are composed, the composite system outputs a continuous, nondeterministic sequence of 1's and 0's. The probability that the composite system will output a 1 at any given state of the system is based on the relative speeds at which the component systems operate. On the other hand, time is not represented in our model. Therefore, we have no way to model the composition of probability-extended state machines in a fully general way.

In future work, we may incorporate the notion of time into the current model. In so doing, it may be possible to incorporate constraints on timing interference (which is not constrained at all in the present work) as well as allow us to properly define the composition of systems and demonstrate the general composability of P-restrictiveness.

In the meantime, we offer the following limited result. In the following sections, the *simple composition* of probability-extended state machines is defined and P-restrictiveness is shown to be composable under simple composition. In defining the simple composition of machines, we assume that the composed machines operate at an identical, constant rate. This is a reasonable assumption in some applications (e.g., two machines executing the same software on the same hardware at the same clock speed).

6.1 The Simple Composition of Systems

Let $A = (S_A, s0_A, E_A, I_A, O_A, T_A)$ and $B = (S_B, s0_B, E_B, I_B, O_B, T_B)$ be two state machines. Provided that $E_A \cap E_B = \emptyset$, we define the simple composition of A and B, denoted $A \| B$, as the machine $(S, s0, E, I, O, T)$, where

$$S = S_A \times S_B$$

$$s0 = (s0_A, s0_B)$$

$$E = E_A \cup E_B$$

$$I = I_A \cup I_B$$

$$O = O_A \cup O_B$$

$$T = \{\, (\langle s_A, s_B\rangle, \gamma, \langle, t_A, t_B\rangle, \tfrac{1}{2}p) \mid \langle s_A, s_B\rangle \in S \text{ and } \langle t_A, t_B\rangle \in S \text{ and } p \in [0,1] \text{ and}$$
$$(((s_A, \gamma, t_A, p) \in T_A \text{ and } s_B = t_B) \text{ or}$$
$$((s_B, \gamma, t_B, p) \in T_B \text{ and } s_A = t_A)\ )\}$$

If it is not true that $E_A \cap E_B = \emptyset$ then $A \| B$ is undefined.

6.2 The Composition of Projections

Let $A = (S_A, s0_A, E_A, I_A, O_A, T_A)$ and $B = (S_B, s0_B, E_B, I_B, O_B, T_B)$ be two state machines. Also let $\langle v_A, \approx_A\rangle$ and $\langle v_B, \approx_B\rangle$ be projections of A and B, respectively. Provided that $v_A \cap v_B = \emptyset$, we define the composite projection, denoted by $\langle v_A, \approx_A\rangle \circ \langle v_B, \approx_B\rangle$, as the view $\langle v, \approx\rangle$, where

$$v = v_A \cup v_B \text{ and}$$

$$(\forall \langle s_A, s_B\rangle, \langle t_A, t_B\rangle \in S)[\langle s_A, s_B\rangle \approx \langle t_A, t_B\rangle \iff s_A \approx_A t_A \text{ and } s_B \approx_B t_B]$$

If it is not true that $v_A \cap v_B = \emptyset$ then the composite projection is undefined.

6.3 The Composability of P-Restrictiveness

Theorem 9: Let $A = (S_A, s0_A, E_A, I_A, O_A, T_A)$ and $B = (S_B, s0_B, E_B, I_B, O_B, T_B)$ be two state machines, and $\langle v_A, \approx_A\rangle$ and $\langle v_B, \approx_B\rangle$ be projections of A and B, respectively. If $A \| B = (S, s0, E, I, O, T)$ is defined, and $\langle v_A, \approx_A\rangle$ is P-restrictive for A and $\langle v_B, \approx_B\rangle$ is P-restrictive for B, then $\langle v_A, \approx_A\rangle \circ \langle v_B, \approx_B\rangle = \langle v, \approx\rangle$ is P-restrictive for $A \| B$.

Proof: Let $\langle A_1, B_1\rangle \in S$ and $\langle A'_1, B'_1\rangle \in S$ be arbitrary states, $\gamma \in E^*$ be an arbitrary event, and $p \in (0,1]$ be a nonzero probability. We must show that

(1) $(\langle A_1, B_1\rangle, \gamma, \langle A'_1, B'_1\rangle, p) \in T$ and $\gamma \in\!\triangleright I$ and $\gamma \notin v \Rightarrow \langle A_1, B_1\rangle \approx \langle A'_1, B'_1\rangle$, and

(2) $\forall \langle A_2, B_2\rangle \in S, \langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle \Rightarrow P_{\langle v, \approx\rangle}(\langle A_1, B_1\rangle, \gamma, \langle A'_1, B'_1\rangle) = P_{\langle v, \approx\rangle}(\langle A_2, B_2\rangle, \gamma, \langle A'_1, B'_1\rangle)$.

To show (1), let $(\langle A_1, B_1\rangle, \gamma, \langle A'_1, B'_1\rangle, p) \in T$ and $\gamma \in\!\triangleright I$ and $\gamma \notin v$. By the definition of T, we have two cases:

Case 1: $(A_1, \gamma, A'_1, 2p) \in T_A$ and $B_1 = B'_1$.
By the definition of v, $\gamma \notin v \Rightarrow \gamma \notin v_A$. Also, since $\gamma \in\!\triangleright I$ and $E_A \cap E_B = \emptyset$ and $\gamma \in E_A^*$, it must be the case that $\gamma \in\!\triangleright I_A$. And so, by the P-restrictiveness of A, $A_1 \approx_A A'_1$.
By the reflexivity of $\approx_B$, $B_1 \approx_B B'_1$ and therefore, $\langle A_1, B_1\rangle \approx \langle A'_1, B'_1\rangle$.

Case 2: $A_1 = A'_1$ and $(B_1, \gamma, B'_1, 2p) \in T_B$.
This case is analagous to Case 1.

Now to show (2), let $\langle A_2, B_2 \rangle$ be a state such that $\langle A_1, B_1 \rangle \approx \langle A_2, B_2 \rangle$.

We must show that $P_{\langle v, \approx \rangle}(\langle A_1, B_1 \rangle, \gamma, \langle A'_1, B'_1 \rangle) = P_{\langle v, \approx \rangle}(\langle A_2, B_2 \rangle, \gamma, \langle A'_1, B'_1 \rangle)$. We will show this in three cases.

Case 1: $\gamma \in v_A$.

$$P_{\langle v, \approx \rangle}(\langle A_1, B_1 \rangle, \gamma, \langle A'_1, B'_1 \rangle) = \sum_{\langle A'_2, B'_2 \rangle \approx \langle A'_1, B'_1 \rangle} p_{(\langle A_1, B_1 \rangle, \gamma, \langle A'_2, B'_2 \rangle)} \qquad [\text{def. } P_{\langle v, \approx \rangle}]$$

$$= \tfrac{1}{2} \sum_{A'_2 \approx A'_1} p_{(A_1, \gamma, A'_2,)} \qquad [\text{def. } T \text{ and } p]$$

$$= \tfrac{1}{2} P_{\langle v_A, \approx_A \rangle}(A_1, \gamma, A'_1) \qquad [\text{def. } P_{\langle v_A, \approx_A \rangle}]$$

$$= \tfrac{1}{2} P_{\langle v_A, \approx_A \rangle}(A_2, \gamma, A'_1) \qquad [\text{P-rest. } \langle v_A, \approx_A \rangle]$$

$$= \tfrac{1}{2} \sum_{A'_2 \approx A'_1} p_{(A_2, \gamma, A'_2,)} \qquad [\text{def. } P_{\langle v_A, \approx_A \rangle}]$$

$$= \sum_{\langle A'_2, B'_2 \rangle \approx \langle A'_1, B'_1 \rangle} p_{(\langle A_2, B_2 \rangle, \gamma, \langle A'_2, B'_2 \rangle)} \qquad [\text{def. } T \text{ and } p]$$

$$= P_{\langle v, \approx \rangle}(\langle A_2, B_2 \rangle, \gamma, \langle A'_1, B'_1 \rangle) \qquad [\text{def. } P_{\langle v, \approx \rangle}]$$

And so, Case 1 holds.

Case 2: $\gamma \in v_B$.
This case is analogous to Case 1.

Case 3: $\gamma \notin v$.

$$P_{\langle v, \approx \rangle}(\langle A_1, B_1 \rangle, \gamma, \langle A'_1, B'_1 \rangle) = \sum_{\substack{\gamma' \in E - v \text{ and} \\ \langle A'_2, B'_2 \rangle \approx \langle A'_1, B'_1 \rangle}} p_{(\langle A_1, B_1 \rangle, \gamma', \langle A'_2, B'_2 \rangle)} \qquad [\text{def. } P_{\langle v, \approx \rangle}]$$

$$= \tfrac{1}{2} \sum_{\substack{\gamma' \in E_A - v_A \text{ and} \\ A_2' \approx A_1'}} p_{(A_1,\gamma',A_2',)} + \tfrac{1}{2} \sum_{\substack{\gamma' \in E_B - v_B \text{ and} \\ B_2' \approx B_1'}} p_{(B_1,\gamma',B_2',)}$$

[def. T and p]

$$= \tfrac{1}{2} P_{\langle v_A, \approx_A \rangle}(A_1, \gamma, A_1') + \tfrac{1}{2} P_{\langle v_B, \approx_B \rangle}(B_1, \gamma, B_1') \qquad \text{[def. } P_{\langle v_A, \approx_A \rangle}\text{]}$$

$$= \tfrac{1}{2} P_{\langle v_A, \approx_A \rangle}(A_2, \gamma, A_1') + \tfrac{1}{2} P_{\langle v_B, \approx_B \rangle}(B_2, \gamma, B_1')$$

[P-rest. $\langle v_A, \approx_A \rangle$ and $\langle v_B, \approx_B \rangle$]

$$= \tfrac{1}{2} \sum_{\substack{\gamma' \in E_A - v_A \text{ and} \\ A_2' \approx A_1'}} p_{(A_2,\gamma',A_2',)} + \tfrac{1}{2} \sum_{\substack{\gamma' \in E_B - v_B \text{ and} \\ B_2' \approx B_1'}} p_{(B_2,\gamma',B_2',)}$$

[def. $P_{\langle v_A, \approx_A \rangle}$ and $P_{\langle v_B, \approx_B \rangle}$]

$$= \sum_{\substack{\gamma' \in E - v \text{ and} \\ \langle A_2', B_2' \rangle \approx \langle A_1', B_1' \rangle}} p_{(\langle A_2, B_2 \rangle, \gamma', \langle A_2', B_2' \rangle)} \qquad \text{[def. } T \text{ and } p\text{]}$$

$$= P_{\langle v, \approx \rangle}(\langle A_2, B_2 \rangle, \gamma, \langle A_1', B_1' \rangle) \qquad \text{[def. } P_{\langle v, \approx \rangle}\text{]}$$

And so, Case 3 holds and the theorem is proved. []

7. Conclusions and Future Work

We have shown with examples that small systems that are restrictive (and that may appear to be reasonable) can contain probabilistic interference (i.e., probabilistic covert channels). Furthermore, it is clear that with larger systems that are shown to be restrictive, probabilistic covert channels may exist that are subtle and difficult to detect. Our extension to McCullough's work provides a security policy that, when applied to a system, guarantees that the system will contain no probabilistic interference.

Additionally, the main example of this report showed how nondeterminism can be used to prevent denial of service, and that useful, nondeterministic systems can be shown to be P-restrictive. Of course, the introduction of nondeterminism to prevent denial of service, as in our example, adversely impacts overall system performance. A tradeoff must be made between prevention of denial of service and system performance.

To apply P-restrictiveness in the development of secure systems, an implementation language that supports the specification of probabilities is needed. The compiler and target machine for this implementation language must accurately implement the specified probabilities, so that the actual system will behave exactly as in the specification, and thus be P-restrictive. Therefore, any effort to apply P-restrictiveness must be a long-term effort.

As discussed in the introduction, our plans for future work are to extend the present model and definition of security to include timing considerations. This will result in a definition of perfect security. Following that, it is our intention to weaken our definition of security to allow a quantifiable amount of

interference. Hopefully, this will make the definition more usable (i.e., more systems will satisfy the definition) and will allow system developers to formally and precisely determine the rate at which a system can leak information. Furthermore, such a definition would allow system designers to trade off the security of the system with other design goals such as system performance and prevention of denial of service.

Acknowledgements

I thank Tom Haigh for a helpful discussion of this work. I especially thank John McLean, Cathy Meadows, and Ira Moskowitz for the many invaluable discussions and technical comments on previous drafts of this work.

REFERENCES

1. J. A. Goguen and J. Meseguer, "Security Policies and Security Models," Proc. of the 1982 IEEE Computer Society Symposium on Computer Security and Privacy, 1982.
2. Joseph A. Goguen and José Meseguer, "Unwinding and Inference Control," Proc. of the 1984 IEEE Computer Society Symposium on Computer Security and Privacy, 1984.
3. Daryl McCullough, "Noninterference and the Composability of Security Properties," Proc. of the 1988 IEEE Computer Society Symposium on Computer Security and Privacy, 1988.
4. David Sutherland, "A Model of Information," Proc. of the 9th National Computer Security Conference, Sept. 1986.
5. Daryl McCullough, "Covert Channels and Degrees of Insecurity", Proc. of the 1988 Workshop on Computer Security Foundations, Franconia, NH, 1988.
6. Todd Fine, J. Thomas Haigh, Richard C. O'Brien, and Dana L. Toups, "Noninterference and Unwinding for LOCK," Proc. of the 1989 Workshop on the Foundations of Computer Security, Franconia, NH, 1989.
7. D. G. Weber, "Quantitative Hook-Up Security for Covert Channel Analysis," Proc. of the 1988 Workshop on the Foundations of Computer Security, Franconia, NH, 1988.
8. David P. Reed and Rajendra K. Kanodia, "Syncronization with Event Counts and Sequencers," *CACM* **22**(2) (1979).
9. Daryl McCullough, "Specifications for Multilevel Security and a Hook-Up Property," Proc. of the 1987 IEEE Computer Society Symposium on Computer Security and Privacy, 1987.
10. J. C. C. White, "Design of a Secure File Management System," MTR-2931, The MITRE Corp., Bedford, MA, June 1974.
11. Thomas H. Hinke and Marvin Schaefer, "Secure Data Management System," TM-(L)-5407/007/00, System Development Corp., Santa Monica, CA, June 1975.
12. Leslie Lamport, "Concurrent Reading and Writing," *CACM* **20**(11) (1977).